

Bounded, minimal, and short representations of unit interval and unit circular-arc graphs

Francisco J. Soulignac*

francisco.soulignac@unq.edu.ar

Abstract

We consider the unrestricted, minimal, and bounded representation problems for unit interval (UIG) and unit circular-arc (UCA) graphs. In the unrestricted version, a proper circular-arc (PCA) model \mathcal{M} is given and the goal is to obtain an equivalent UCA model \mathcal{U} . We show a linear time algorithm with negative certification that can also be implemented to run in logspace. In the bounded version, \mathcal{M} is given together with some lower and upper bounds that the beginning points of \mathcal{U} must satisfy. We develop a linear space $O(n^2)$ time algorithm for this problem. Finally, in the minimal version, the circumference of the circle and the length of the arcs in \mathcal{U} must be simultaneously as minimum as possible. We prove that every UCA graph admits such a minimal model, and give a polynomial time algorithm to find it. We also consider the minimal representation problem for UIG graphs. As a bad result, we show that the previous linear time algorithm fails to provide a minimal model for some input graphs. We fix this algorithm but, unfortunately, it runs in linear space $O(n^2)$ time. Finally, we apply the minimal representation algorithms so as to find the minimum powers of paths and cycles that contain a given UIG and UCA models, respectively.

Keywords: unit circular-arc graphs, unit interval graphs, recognition problem, bounded representation problem, minimal model, powers of paths and cycles.

1 Introduction

In this article we are concerned with some recognition and representation problems for unit interval and unit circular-arc graphs. A *proper circular-arc (PCA) model* is a pair $\mathcal{M} = (C, \mathcal{A})$ where C is a circle and \mathcal{A} is a family of inclusion-free arcs of C in which no pair of arcs in \mathcal{A} cover C . If some point of C is crossed by no arcs, then \mathcal{M} is a *proper interval (PIG) model*. *Unit circular-arc (UCA)* and *unit interval (UIG)* models correspond to the PCA and PIG models in which all the arcs have the same length, respectively. Every PCA model \mathcal{M} is associated with a graph $G(\mathcal{M})$ that contains a vertex for each of its arcs, where two vertices are adjacent if and only if their corresponding arcs have a nonempty intersection. A graph G is a *proper circular-arc (PCA) graph* when it is isomorphic to $G(\mathcal{M})$ for some PCA model \mathcal{M} . In such a case, G is said to *admit* the model \mathcal{M} , while \mathcal{M} is said to *represent* G . *Proper interval (PIG)*, *unit circular-arc (UCA)*, and *unit interval (UIG)* graphs are defined analogously.

The recognition problem is well solved for UIG graphs. Indeed, Roberts' PIG=UIG Theorem states that every PIG graph admits a UIG model [31]. Hence, it suffices to determine if G is a PIG graph, a task that can be accomplished in linear time (e.g. [13]) or logspace [20]. Moreover, there are *certifying* algorithms that exhibit either a PIG model or a forbidden induced subgraph according to whether the input graph is PIG or not.

*CONICET and Departamento de Ciencia y Tecnología, Universidad Nacional de Quilmes, Bernal, Argentina.

Knowing that G is a UIG graph tells us nothing about its UIG models. In this article we deal with the stronger (*unrestricted*) *representation* (REP) problem in which a UIG model \mathcal{U} equivalent to an input PIG model \mathcal{M} is to be found. By *equivalent* we mean that the extremes of \mathcal{U} must appear in the same order as in \mathcal{M} . The representation problem can be generalized to the *partial representation extension* (REPEXT) problem in which some arcs of \mathcal{M} are *pre-drawn*, and \mathcal{U} must contain these arcs. REPEXT is in turn a special case of the more general *bounded representation* (BOUNDREP) problem in which a length $\ell \in \mathbb{Q}$ is given together with lower and upper bounds $d_\ell(A), d_r(A) \in \mathbb{Q}$ for each arc A of \mathcal{M} , and the goal is to produce a UIG model \mathcal{U} in which all the arcs have length ℓ in such a way that $d_\ell(A) \leq s(A) \leq d_r(A)$ for every arc A . Here $s(A) \in \mathbb{Q}$ represents the beginning point of A . In this article we consider a further generalization of BOUNDREP in which $\ell, d_\ell(A), d_r(A)$ are integers, and each beginning point $s(A)$ of \mathcal{U} is required to be an integer as well. We refer to this problem as the INTBOUNDREP; as far as we know, INTBOUNDREP has not been considered before.

REP is a classical problem whose research is even older than PIG graphs. Indeed, REP is one of the motivations in the pioneering philosophical work by Goodman [12], which dates back to the 1940's. Moreover, Fine and Harrop [9] developed, in 1957, an effective method to transform a weak mapping of an array (i.e., a PIG model) into a uniform mapping of the same array (i.e., a PIG model of a power of a path); this algorithm is actually the first proof of Robert's PIG=UIG theorem, as far as our knowledge extends. Linear time algorithms for REP are known since more than two decades [6, 25, 27] and, recently, a logspace implementation has been devised [21].

The research on REPEXT and BOUNDREP did not begin until recently and, consequently, they are not as studied as REP. We remark that these problems are defined not only for UIG graphs, but for several graph classes with geometric representations. In the last few years, the partial representation extension and the bounded representation problems were studied for several graph classes [1–4, 16, 17, 19]. Concerning PIG graphs, Balko et al. [1] show that the bounded representation problem is solvable in $O(n^2)$ time. Regarding UIG graphs, Klavík et al. [18, 19] designed an $O(n^2 + nD)$ time algorithm for BOUNDREP, where D is the cost of multiplying large numbers (requiring r bits, where r is the total space consumed by the bounds). As the main open problem, the authors inquire if there exists an algorithm running in less than $O(n^2 + nD)$ time. In [18, 19], a generalization of BOUNDREP in which the output UIG model \mathcal{U} needs not be equivalent to the input PIG model \mathcal{M} is also considered; what the authors ask is for $G(\mathcal{U})$ to be isomorphic to $G(\mathcal{M})$. Whereas BOUNDREP is polynomial, this generalization is NP-complete [18, 19].

While introducing their research on REPEXT, Klavík et al. state that “specific properties of unit interval representations were **never** investigated since it is easier to work with combinatorially equivalent proper interval representations” [18]. However, in 1990, Pirlot proved that every PIG graph admits a *minimal* UIG model [28]. Tough Pirlot's work is not of an algorithmic nature, the main tool he uses is a space efficient representation of PIG models called the *synthetic graph*. With the aid of an appropriate weighing, this graph reflects the separation constraints that all the equivalent UIG models must satisfy. As part of his work, Pirlot solves the problem of determining if a PIG graph admits a UIG model in which all the arcs have integer endpoints and a given length ℓ . Clearly, this is a specific property of UIG models. Moreover, Pirlot introduces synthetic graphs to solve the linear program in [19, Proposition 5.4] (except for the bound constraints) and, vice versa, the graph used in [19, Proposition 5.4] is a synthetic graph (plus two vertices for modeling the bounds).

Similarly as above, in [10] Gardi claimed that, up to 2007, the algorithm by Corneil et al. [6] was the **only** one able to solve REP in linear time. Again, by Pirlot's theorem, it makes sense to consider the *minimal UIG representation* (MINUIG) problem, in which an input PIG model

has to be transformed into an equivalent minimal UIG model. By taking a deeper look to synthetic graphs, Mitas [27] devised a linear time algorithm to solve MINUIG and, thus, REP. In the present manuscript we show that Mitas' algorithm sometimes fails to find the minimal model. Yet, her algorithm correctly solves REP in linear time. We remark that Mitas' (1994) algorithm is contemporary to the one by Corneil et al. (1995).

MINUIG is implicitly solved in a recent article by Costa et al. [7], where the authors devise an $O(n^2)$ time and space algorithm to solve the MINP_q^k problem. In the MINP_q^k problem we are given a PIG model \mathcal{M} and the goal is to find a UIG model \mathcal{U} representing a power of a path P_q^k in such a way that \mathcal{M} is equivalent to some induced subgraph \mathcal{U}' of \mathcal{U} and q, k are as minimum as possible. As proven in [9, 26], MINP_q^k is always solvable. Moreover, \mathcal{U} needs not be explicitly constructed, as it is implied by \mathcal{U}' . In fact, \mathcal{U}' is the solution to MINUIG, as it follows from [26] (see also Section 9). In [32, Chapter 9], Soulignac mentions that Mitas' algorithm can be used to find \mathcal{U}' in linear time. Yet, Costa et al. do not mention this fact in [7] although they reference [32] to explain the strong relation between REP and MINP_q^k .

In this article we consider the unrestricted, bounded, and minimal representation problems for the broader class of unit circular-arc graphs. As far as our knowledge extends, only the unrestricted version has been considered, while Lin and Szwarcfiter leave some open problems related to the minimal representation problem [24].

As for PIG graphs, the recognition problem for PCA graphs is solvable in linear time [15, 33] or logspace [21]. Again, a PCA model or a forbidden induced subgraph is obtained according to whether the input graph is PCA or not. We remark, however, that solving the recognition problem for PCA graphs is not enough to solve the recognition problem for UCA graphs, as not every PCA graph is UCA. In 1974, Tucker showed a characterization by forbidden subgraphs of those PCA graphs that are UCA [35]. His proof yields an effective method to transform a PCA model \mathcal{M} into an equivalent UCA model \mathcal{U} . Unfortunately, the extremes of \mathcal{U} are not guaranteed to be of a polynomial size and, thus, the corresponding representation algorithm cannot be regarded as polynomial. More than three decades later, in 2006, Durán et al. [8] described how to obtain a forbidden subgraph in $O(n^2)$ time, thus solving the recognition problem. The representation problem remained unsolved until Lin and Szwarcfiter showed how to transform any PCA model into an equivalent UCA model in linear time [24]. Their algorithm, however, does not output a *negative* certificate when the input graph is not UCA. The problem of finding a forbidden subgraph in linear time was solved by Kaplan and Nussbaum in [15]. Yet, up to this date, there is no *unified* algorithm for solving the transformation problem while providing a negative certificate when the input model has no equivalent UCA models. In [22], Köbler et al. mention that the representation problem in logspace is still open.

1.1 Contributions and outline

Synthetic graphs appeared more than two decades ago, and they are covered in detail in a book by Pirlot and Vincke [30, Chapter 4]. Pirlot and Mitas' articles are written in terms of semiorders; their emphasis is on preference modeling and order theory. This could be, perhaps, the reason why synthetic graphs have gone unnoticed for many researchers in the field of algorithmic graph theory. In this manuscript we generalize synthetic graphs to PCA models and we apply them to solve REP, (INT)BOUNDREP, MINUIG (and its generalization MINUCA), and MINP_q^k (and its generalization MINC_q^k) for UCA graphs. One of our goals is to show that synthetic graphs provide a simpler theoretical ground for understanding PCA models with separation constraints. For this reason, we re-prove some known theorems or rewrite some known algorithms in terms of synthetic graphs.

The manuscript is organized as follows. In Section 2 we describe the terminology employed.

In Section 3 we introduce synthetic graphs and show how to use them to solve BOUNDERP and INTBOUNDERP in $O(n^2)$ time, improving over the algorithm in [18, 19] even when restricted to UIG graphs. In Section 4 we show a new version of Tucker’s characterization which implies a linear time representation algorithm with negative certification, thus solving the problem posed in [15]. The implementation of this algorithm appears in Section 6, while Section 6.1 contains a logspace implementation that solves the open problem of [22]. To apply our algorithm we need to find, as we call it, the ratio of the input model. This ratio can be computed by invoking the recognition algorithm of [15]. However, we show an alternative implementation in Section 5, by taking advantage of synthetic graphs. The forbidden structure that we employ to characterize UCA graphs is a cycle of the synthetic graph which, a priori, is unrelated to the (a, b) -independents and (x, y) -circuits employed by Tucker. In Section 7 we show that, in fact, these structures are strongly related. In Section 8 we extend the concept of minimal models to UCA graphs, and prove that every UCA graph admits a minimal model. An algorithm to generate such a model in polynomial time is also exhibited. In Section 8.1, we consider the MINUIG problem. We show that, even though Mitas’ algorithm correctly solves REP, it sometimes fails to provide a minimal model. We propose a patch but, unfortunately, the new algorithm runs in $O(n^2)$ time. In Section 9 we show how MINUIG and MINUCA can be used so as to solve MINP_q^k and MINC_q^k , respectively. The obtained algorithm for MINP_q^k runs in $O(n^2)$ time but it consumes only linear space. Finally, we include some further remarks and open problems in Section 10.

1.2 What is linear time for PCA models?

As discussed in [32], every PCA model \mathcal{M} can be encoded with $O(n)$ bits, n being the number of arcs in \mathcal{M} . Thus, in theory, an algorithm on \mathcal{M} is linear when it applies $O(n)$ operations on bits. However, it is a common practice to assume that \mathcal{M} is implemented with $\Theta(n)$ pointers in such a way that the extremes of an arc can be obtained in $O(1)$ time when the other extreme is given (see [32]). Following this tradition, we state that an algorithm is linear when it performs $O(n)$ operations on pointers of size $\Theta(\log n)$.

2 Preliminaries

In this article we consider simple (undirected) graphs and multidigraphs with no loops. For the sake of simplicity, we refer to the latter as *digraphs* and to its directed edges as *edges*, unless otherwise stated. For a (di)graph G we write $V(G)$ and $E(G)$ to denote the sets of vertices and bag of edges of G , respectively, while we use n and m to denote $|V(G)|$ and $|E(G)|$, respectively. For any pair $u, v \in V(G)$, we denote the (directed) edge between u and v (*from or starting at u to or ending at v*) by uv . This notation is used regardless of whether uv is an edge of G or not. To avoid confusions, we write $u \rightarrow v$ as an equivalent of uv when G is a digraph; the *in-* and *out-degrees* of v are the number of vertices u such that $u \rightarrow v$ and $v \rightarrow u$, respectively.

A walk W of a (di)graph G is a sequence of vertices v_1, \dots, v_k such that $v_i v_{i+1}$ is an edge of G , for every $1 \leq i < k$. Walk W goes *from (or starts at) v_1 to (or ends at) v_k* . Each walk can be regarded as the bag of edges $\{v_i v_{i+1} \mid 1 \leq i < k\}$. For the sake of simplicity, we make no distinctions about W begin a sequence of vertices or a bag of edges. We say that W is a circuit when $v_k = v_1$, that W is a *path* when $v_i \neq v_j$ for every $1 \leq i < j \leq k$, and that W is a *cycle* when it is a circuit and v_1, \dots, v_{k-1} is a path. Sometimes we also say that W is a circuit when $v_1 \neq v_k$, to mean that W, v_1 is a circuit. If G contains no cycles, then G is an *acyclic* digraph.

An *edge weighing*, or simply a *weighing*, of a (di)graph G is a function $w: E(G) \rightarrow \mathbb{R}$. The value $w(uv)$ is referred to as the *weight* of uv (with respect to w). For any bag of

edges, the *weight* of E (with respect to an edge weighing w) is $w(E) = \sum_{uv \in E} w(uv)$. We use two distance measures on a (di)graph G with a weighing w . For $u, v \in V(G)$, we denote by $\mathbf{d}^*w(G, u, v)$ the maximum among the weights of the walks from u to v , while $\mathbf{d}w(G, u, v)$ denotes the maximum among the weights of the paths starting at u and ending at v . Note that $\mathbf{d}w(G, u, v) < \infty$ for every u, v , while $\mathbf{d}^*w(G, u, v) = \mathbf{d}w(G, u, v)$ when G contains no cycle of positive weight [5]. For a weighing w' , we write $(\mathbf{d}w \circ \mathbf{d}w')(G, u, v) = \max\{w(W) \mid W \text{ is a path from } u \text{ to } v \text{ with } w'(W) = \mathbf{d}w'(G, u, v)\}$. In other words, $\mathbf{d}w \circ \mathbf{d}w'$ measures the w -distance from u to v when only those walks that impose the maximum w' -distance from u to v are considered. For the sake of notation, we omit the parameter G when there are no ambiguities.

A *straight plane* (di)graph, or simply a *plane* (di)graph, is a (di)graph whose vertices are coordinates in the plane and whose edges are non-crossing straight lines. Similarly, a *toroidal* (di)graph is a (di)graph whose vertices and edges can be placed on the surface of a torus in such a way that no pair of edges intersect.

A *proper circular-arc* (PCA) model \mathcal{M} is a pair (C, \mathcal{A}) , where C is a circle and \mathcal{A} is a collection of inclusion-free arcs of C such that no pair of arcs in \mathcal{A} cover C . When traversing the circle C , we always choose the clockwise direction. If s, t are points of C , we write (s, t) to mean the arc of C defined by traversing the circle from s to t ; s and t are the *extremes* of (s, t) , while s is the *beginning point* and t the *ending point*. For $A \in \mathcal{A}$, we write $A = (s(A), t(A))$. The *extremes* of \mathcal{A} are those of all arcs in \mathcal{A} , and two extremes s_1s_2 of \mathcal{M} are *consecutive* when there is no extreme $s \in (s_1, s_2)$ (note that s_2s_1 is not consecutive in this case). We assume C has a special point 0 that is used for describing the bounds on the extremes (cf. below). This point is only denotational for the unbounded case. For every pair of points p_1, p_2 , we write $p_1 < p_2$ to indicate that p_1 appears before p_2 in a traversal of C from 0. Similarly, we write $A_1 < A_2$ to mean that $s(A_1) < s(A_2)$ for any pair of arcs A_1, A_2 on C .

A *unit circular-arc* (UCA) model is a circular-arc model \mathcal{M} in which all the arcs have the same length. Let $A_1 < \dots < A_n$ be the arcs of $\mathcal{M} = (C, \mathcal{A})$, $c, \ell \in \mathbb{Q}_{>0}$, $d, d_s \in \mathbb{Q}_{\geq 0}$, and $d_\ell, d_r: \mathcal{A} \rightarrow \mathbb{Q}_{\geq 0}$. We say that \mathcal{M} is a $(c, \ell, d, d_s, d_\ell, d_r)$ -CA model when:

(unit₁) C has circumference c ,

(unit₂) all the arcs of \mathcal{A} have length ℓ ,

(unit₃) (p_1, p_2) has length at least d for every pair of consecutive extremes p_1p_2 ,

(unit₄) (s_1, s_2) has length at least $d + d_s$ for any pair of beginning points s_1, s_2 , and

(unit₅) $d_\ell(A_i) \leq s(A_i) \leq c - d_r(A_i)$ for every $1 \leq i \leq n$.

Intuitively, \mathcal{M} is a UCA model in which the extremes are separated by at least d space, the beginning points are separated by $d + d_s$ space, and $d_\ell(A_i)$ and $d_r(A_i)$ are lower bounds of the separation from 0 to $s(A_i)$ and from $s(A_i)$ to 0, respectively. We simply write that \mathcal{M} is a (c, ℓ, d, d_s) -CA model to indicate that $d_\ell = d_r = 0$, and that \mathcal{M} is a (c, ℓ) -CA model to mean that \mathcal{M} is a $(c, \ell, 1, 0)$ -CA model. To further simplify the notation, we refer to the tuple $u = (c, \ell, d, d_s, d_\ell, d_r)$ as a *UCA descriptor*, and we say that u is *integer* when c, ℓ, d, d_s, d_ℓ , and d_r are integers. Similarly, a u -CA model \mathcal{M} is *integer* when c, ℓ and all the extremes of \mathcal{M} are integers.

A *proper interval* (PIG) model is a PCA model \mathcal{M} in which no arc crosses 0; if \mathcal{M} is also UCA, then \mathcal{M} is a *unit interval* (UIG) model. Any UIG model \mathcal{M} is a u -CA model for some large enough c ; for simplicity, we just write $c = \infty$ in this case. For this reason, we say that \mathcal{M} is an $(\ell, d, d_s, d_\ell, d_r)$ -IG (resp. (ℓ, d, d_s) , ℓ -IG) model when \mathcal{M} is a $(\infty, \ell, d, d_s, d_\ell, d_r)$ -CA

(resp. (∞, ℓ, d, d_s) , (∞, ℓ) -CA) model. That is, \mathcal{M} is an $(\ell, d, d_s, d_\ell, d_r)$ -IG model when all the arcs have length ℓ , every pair of consecutive extremes is separated by d space, every pair of beginning points is separated by $d + d_s$ space, and d_ℓ and d_r impose lower and upper bounds on the beginning points of \mathcal{M} .

Each PCA model \mathcal{M} represents a *proper circular-arc* graph $G(\mathcal{M})$ that contains a vertex for each arc of \mathcal{M} where two vertices are adjacent if and only if their corresponding arcs have nonempty intersection. Conversely, we say that a graph G admits a PCA model \mathcal{M} to mean that G is isomorphic to $G(\mathcal{M})$. If \mathcal{M} is UCA, then $G(\mathcal{M})$ is a *unit circular-arc* (UCA) graph, while if \mathcal{M} is PIG (resp. UIG), then $G(\mathcal{M})$ is a *proper interval* (PIG) (resp. *unit interval*; UIG) graph.

Clearly, two PCA models $\mathcal{M}_1 = (C_1, \mathcal{A}_1)$ and $\mathcal{M}_2 = (C_2, \mathcal{A}_2)$ are equal when $C_1 = C_2$ and $\mathcal{M}_1 = \mathcal{M}_2$. We say that \mathcal{M}_1 is *equivalent* to \mathcal{M}_2 when the extremes of \mathcal{M}_1 appear in the same order as in \mathcal{M}_2 . Formally, \mathcal{M}_1 and \mathcal{M}_2 are equivalent if there exists $f: \mathcal{A}_1 \rightarrow \mathcal{A}_2$ such that $e(f(A))e'(f(B))$ are consecutive if and only if $e(A)e'(B)$ are consecutive, for $e, e' \in \{s, t\}$. By definition, \mathcal{M}_1 and \mathcal{M}_2 are equivalent whenever they are equal.

In this manuscript we consider several related recognition problems. In the *representation* (REP) problem a UCA model equivalent to an input PCA model \mathcal{M} must be generated. Of course, REP is unsolvable when \mathcal{M} is equivalent to no UCA model, a *negative certificate* is desired in such a case. In the *u*-REP problem, a (an integer) UCA descriptor u is given together with \mathcal{M} , and the goal is to build a (an integer) *u*-CA model \mathcal{U} . We remark that an integer \mathcal{U} equivalent to \mathcal{M} exists whenever u is integer and *u*-REP is solvable. The *bounded representation* (BOUNDREP) is a slight variation of *u*-REP in which a feasible $d > 0$ must be found by the algorithm, as it is not given as input. That is, we are given a PCA model $\mathcal{M} = (C, \mathcal{A})$ together with $c, \ell \in \mathbb{Q}_{>0}$, $d_s \in \mathbb{Q}_{\geq 0}$ and $d_\ell, d_r: \mathcal{A} \rightarrow \mathbb{Q}_{\geq 0}$, and we ought to find a *u*-CA model equivalent to \mathcal{M} for some UCA descriptor $u = (c, \ell, d, d_s, d_\ell, d_r)$ with $d \in \mathbb{Q}_{>0}$. The *integer bounded representation* (INTBOUNDREP) problem is a generalization of BOUNDREP in which all the input values are integers and the output model must be integer as well. We also study the MINUIG, MINUCA, MINP_q^k , and MINC_q^k problems that are related to *minimal* models. We postpone their definitions to Sections 8 and 9.

2.1 Restrictions on the input models

As it is customary in the literature, in this article we assume that all the arcs of a PCA model \mathcal{M} are open and no two extremes of \mathcal{M} coincide. The reason behind these assumptions is that \mathcal{M} can always be transformed into an equivalent model \mathcal{M}' that satisfies these properties. A word of caution is required, though, as in this article we deal with the lengths of the arcs. If we allow coincidences in the extremes of \mathcal{M} , for instance, it is possible to shrink the length of the arcs or the circle of some UCA models. We emphasize, nevertheless, that all the arguments in this article, with the obvious adjustments, work equally well without these assumptions. In particular, note that the articles by Klavík et al., Mitás, and Pirlot allow coincident extremes [18, 27–29].

By definition, for us PCA models cannot have two arcs covering the circle. This is a somehow artificial restriction that we impose for the sake of simplicity. In general, this class of models is said to be *normal*. However, it is well known that every non-normal PCA model can be transformed into a normal PCA model in linear time or logspace (see e.g. [15]). Moreover, note that if two arcs in a UIG model cover the circle, then such a model represents a complete graph. The complete graph on n vertices admits the *minimal* UIG model $\{(i, i + n + 1) \mid 1 \leq i \leq n\}$, thus we do not lose much by excluding these non-normal models when dealing with REP, MINUCA, and MINUIG. In turn, the fact that \mathcal{M} is normal is not used in Theorem 1, thus

(INT)BOUNDREP is also solvable for non-normal models.

Finally, we require two additional restrictions on the input PCA models for technical reasons. We say that a PCA model \mathcal{M} with arcs $A_1 < \dots < A_n$ is *trivial* when either

1. $s(A_n) < t(A_1)$, or
2. $s(A_i)t(A_i)$ are consecutive for some $1 \leq i \leq n$.

If 1. holds, then we cannot claim that $h(\mathcal{M}) \geq 1$ in Section 3.2. However, in this case \mathcal{M} represents a complete graph and $\{(i, i + n + 1) \mid 1 \leq i \leq n\}$ is the unique minimal and integer UCA model equivalent to \mathcal{M} . Thus all the considered problems are trivial in this case. If 2. is true, then $A_i \rightarrow A_i$ is a loop of the digraph $\mathcal{B}(\mathcal{M})$ defined in Section 3. We can certainly allow the existence of such a loop in \mathcal{B} . However, this edge plays no role in the considered problems as $\text{sep}(A_i \rightarrow A_i) < 0$ by (sep₃).

3 The synthetic graph of a PCA model

Pirlot introduced the synthetic graph of a PIG model [28, 29] to represent the separation constraints of its extremes in any equivalent UIG representation. In this section we extend them to PCA models and we show that they correctly reflect the separation constraints in any equivalent UCA model.

Let $\mathcal{M} = (C, \mathcal{A})$ be a PCA model with arcs $A_1 < \dots < A_n$. The *bounded synthetic graph* of \mathcal{M} is the digraph $\mathcal{B}(\mathcal{M})$ (see Figure 1) that has a vertex $v(A_i)$ for each $A_i \in \mathcal{A}$ and a vertex A_0 , and whose edge set is $E_\sigma \cup E_\nu \cup E_\eta \cup E_\beta$, where:

- $E_\sigma = \{v(A_i) \rightarrow v(A_{i+1}) \mid 1 \leq i \leq n, A_{n+1} = A_1\}$,
- $E_\nu = \{v(A_i) \rightarrow v(A_j) \mid t(A_i)s(A_j) \text{ are consecutive in } \mathcal{M}\}$,
- $E_\eta = \{v(A_i) \rightarrow v(A_j) \mid s(A_i)t(A_j) \text{ are consecutive in } \mathcal{M}\}$, and
- $E_\beta = \{A_0 \rightarrow v(A_i), v(A_i) \rightarrow A_0 \mid 1 \leq i \leq n\}$.

The edges in E_σ , E_ν , E_η , and E_β are said to be the *steps*, *noses*, *hollows*, and *bounds* of $\mathcal{B}(\mathcal{M})$, respectively. (Note that E_σ , E_ν and E_η could have a nonempty intersection, even if this is not the common case. However, $\mathcal{B}(\mathcal{M})$ has no loops as \mathcal{M} is not trivial.) For the sake of simplicity, we usually drop the parameter \mathcal{M} from $\mathcal{B}(\mathcal{M})$ when no ambiguities are possible. Moreover, we regard the arcs of \mathcal{M} as being the vertices of \mathcal{B} , thus we may say that $A_i \rightarrow A_j$ is a nose instead of writing that $v(A_i) \rightarrow v(A_j)$ is a nose.

We classify the edges of \mathcal{B} in two classes according to the positions of their arcs. We say a step (resp. nose) $A_i \rightarrow A_j$ is *internal* when $i < j$, while a hollow is *internal* when $i > j$. Non-internal edges are referred to as *external*; in particular, all the bounds are external. Observe that every step is internal except $A_n \rightarrow A_1$. Similarly, a nose $A_i \rightarrow A_j$ is internal if and only if the arc $(s(A_i), s(A_j))$ does not cross 0, while a hollow $A_i \rightarrow A_j$ is internal if and only if $(s(A_j), s(A_i))$ does not cross 0. Since the purpose of A_0 is to represent the point 0 of \mathcal{M} , we can say, in short, that $A_i \rightarrow A_j$ is internal when 0 is not crossed in the traversal of the extremes involved in the definition of $A_i \rightarrow A_j$.

We define a special edge weighing sep_u of \mathcal{B} whose purpose is to indicate how far or close must $s(A_i)$ and $s(A_j)$ be in any u -CA model equivalent to \mathcal{M} , for every edge $A_i \rightarrow A_j$ of \mathcal{B} . For a UCA descriptor u , the edge weighing sep_u is such that:

$$(\text{sep}_1) \quad \text{sep}_u(A_i \rightarrow A_j) = d + d_s - cq \text{ if } A_i \rightarrow A_j \text{ is a step,}$$

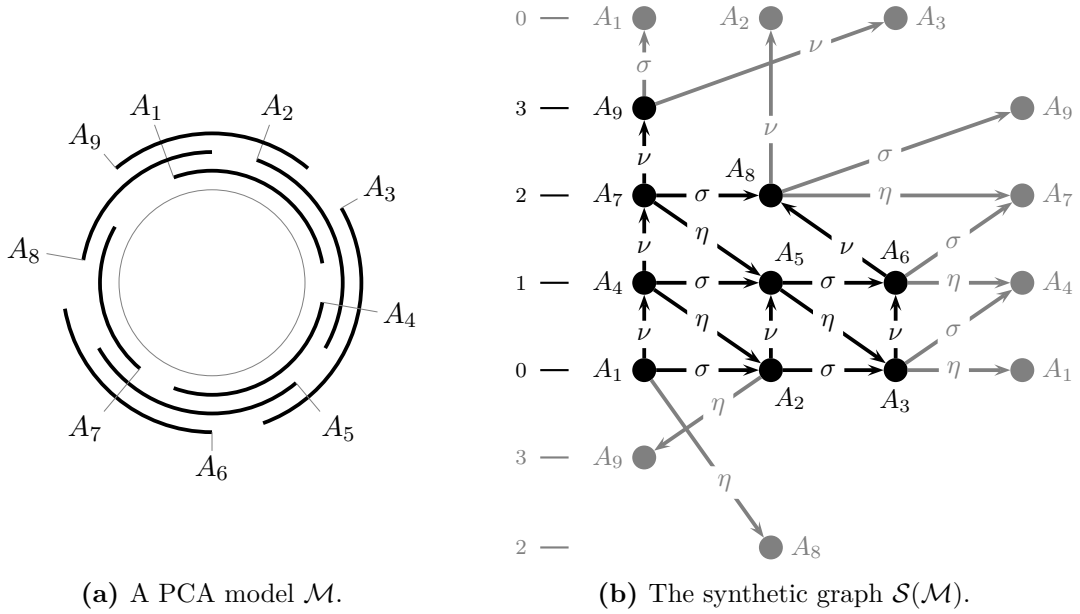


FIGURE 1. Synthetic graph $\mathcal{B} \setminus A_0$ of a PCA model. Each gray vertex corresponds to a black vertex (we separate them for the sake of exposition) and each edge is drawn only once. The edges are labeled with ν , η , and σ according to whether they are noses, hollows, and steps, respectively. The height of \mathcal{M} is $h = 3$ and each vertex is drawn in a row that corresponds to its height; the height is indicated to the left. Note that there are 1 -, $(-h)$ -, and $(1-h)$ -noses, 0 -, (-1) -, h -, and $(h-1)$ -hollows, and 0 -, 1 -, and $(-h)$ -steps.

(sep₂) $\text{sep}_u(A_i \rightarrow A_j) = d + \ell - cq$ if $A_i \rightarrow A_j$ is a nose,

(sep₃) $\text{sep}_u(A_i \rightarrow A_j) = d + cq - \ell$ if $A_i \rightarrow A_j$ is a hollow, and

(sep₄) $\text{sep}_u(A_0 \rightarrow A_i) = d_\ell(A_i)$ and $\text{sep}(A_i \rightarrow A_0) = d_r(A_i) - c$ for every $1 \leq i \leq n$,

where $q \in \{0, 1\}$ equals 0 if and only if $A_i \rightarrow A_j$ is internal. For the sake of notation, we omit the subscript u from sep when no ambiguities are possible. Suppose for a moment that \mathcal{M} is a u -CA model. By definition, $s(A_j) \geq s(A_i) + \ell + d$ when $A_i \rightarrow A_j$ is an internal nose of \mathcal{B} , while $s(A_j) \geq s(A_i) + \ell + d - c$ when $A_i \rightarrow A_j$ is an external nose of \mathcal{B} . Thus, equation (sep₂) models the non-intersection constraints imposed by the noses of \mathcal{B} . A similar analysis shows that (sep₁) indicates that all the beginning points must be at distance at least $d + d_s$, (sep₃) models the intersection constraints imposed by the hollows of \mathcal{B} , and (sep₄) models the bound constraints, assuming that A_0 represents 0 in \mathcal{M} .

As we shall see in Theorem 1, a u -CA model equivalent to \mathcal{M} exists when the longest path problem with weight sep has a feasible solution on \mathcal{B} . In such case, a u -CA model can be generated by observing the distances from A_0 . With this in mind, we define $\mathcal{U}(\mathcal{M}, u)$ to be the u -CA model with arcs U_1, \dots, U_n such that $s(U_i) = \mathbf{dsep}(A_0, A_i)$, for every $1 \leq i \leq n$ (we assume arithmetic modulo c). For simplicity, we omit \mathcal{M} and u from \mathcal{U} as usual.

Theorem 1. *The following statements are equivalent for a PCA model \mathcal{M} with arcs $A_1 < \dots < A_n$ and a (an integer) UCA descriptor u :*

- (i) \mathcal{M} is equivalent to a u -CA model.
- (ii) $\text{sep}(\mathcal{W}) \leq 0$ for every cycle \mathcal{W} of \mathcal{B} .
- (iii) \mathcal{U} is a (an integer) u -CA model equivalent to \mathcal{M} .

Proof. (i) \Rightarrow (ii). Suppose \mathcal{M} is equivalent to a u -CA model \mathcal{M}' with arcs A'_1, \dots, A'_n such that A'_i corresponds to A_i for $1 \leq i \leq n$. Write $s(A'_0)$ to mean the point 0 of \mathcal{M}' . Then, it is not hard to see (cf. above) that $s(A'_j) \geq s(A'_i) + \text{sep}(A_i \rightarrow A_j)$ for every edge $A_i \rightarrow A_j$ of \mathcal{B} . Hence, by induction, $s(A'_i) \geq s(A'_i) + \text{sep}(\mathcal{W})$ for every cycle \mathcal{W} of \mathcal{B} that contains A_i .

(ii) \Rightarrow (iii). Let $U_1 < \dots < U_n$ be the arcs of \mathcal{U} , $U_{n+1} = U_1$, $A_{n+1} = A_1$, and note that $\mathbf{d}^*\text{sep}(A_i, A_j) = \mathbf{d}\text{sep}(A_i, A_j)$ for every $0 \leq i, j \leq n$ as \mathcal{B} has no cycles of positive length. Thus, by (sep₄), \mathcal{U} satisfies (unit₅) as $s(U_i) = \mathbf{d}\text{sep}(A_0, A_i) \geq d_\ell(A_i)$ and $s(U_i) + d_r(A_i) - c = \mathbf{d}\text{sep}(A_0, A_i) + d_r(A_i) - c \leq \mathbf{d}\text{sep}(A_0, A_0) = 0$. Since \mathcal{U} satisfies (unit₁)–(unit₂) by definition, it follows that \mathcal{U} is a $(c, \ell, d', d'_s, d_\ell, d_r)$ -CA model for some d', d'_s . To prove that \mathcal{U} is a u -CA model equivalent to \mathcal{M} , it suffices to see that (a) $s(U_i) + d + d_s \leq s(U_{i+1})$ for every $1 \leq i \leq n$, (b) $s(U_i) + d \leq t(U_j)$ when $s(A_i)t(A_j)$ are consecutive in \mathcal{M} , and (c) $t(U_i) + d \leq s(U_j)$ when $t(A_i)s(A_j)$ are consecutive in \mathcal{M} .

- (a) $A_i \rightarrow A_{i+1}$ is a step, thus $s(U_i) + d + d_s = \mathbf{d}\text{sep}(A_0, A_i) + d + d_s \leq \mathbf{d}\text{sep}(A_0, A_{i+1}) = s(U_{i+1})$.
- (b) $A_i \rightarrow A_j$ is a hollow of \mathcal{B} ; let $q \in \{0, 1\}$ be 1 if and only if $(s(A_j), s(A_i))$ crosses 0. Note that, equivalently, $q = 1$ if and only if $A_i \rightarrow A_j$ is external. Thus, $t(U_j) = s(U_j) + \ell - cq = \mathbf{d}^*\text{sep}(A_0, A_j) + \ell - cq \geq \mathbf{d}^*\text{sep}(A_0, A_i) + \text{sep}(A_i \rightarrow A_j) + \ell - cq = s(U_i) + d$.
- (c) $A_i \rightarrow A_j$ is a nose of \mathcal{B} ; if $q \in \{0, 1\}$ equals 1 when $(s(A_i), s(A_j))$ crosses 0, then $s(U_j) \geq \mathbf{d}^*\text{sep}(A_0, A_i) + 1 + \ell - cq \geq t(U_i) + d$.

(iii) \Rightarrow (i). Trivial. □

When restricted to PIG models, Theorem 1 is a somehow alternative formulation of Proposition 2.5 in [28]; see also Proposition 5.4 in [18].

3.1 The bounded representation problem

Though simple enough, Theorem 1 allows us to solve u -REP as follows. First, we build the digraph \mathcal{B} in which every edge $A_i \rightarrow A_j$ is weighed with $s_{ij} = \text{sep}_u(A_i, A_j)$. Then, we invoke the Bellman-Ford shortest path algorithm [5] on \mathcal{B} to obtain $s_i = \mathbf{d}^*\text{sep}(A_0, A_i)$ for every $0 \leq i \leq n$. If Bellman-Ford ends in success, then we output $\mathcal{U}(\mathcal{M}, u)$; otherwise, we output the cycle of positive weight found as the negative certificate.

Bellman-Ford computes each value $s_i = s_i^n$ in an iterative manner. At iteration k , the value of s_i is updated to $s_i^k = \max\{s_j^{k-1} + s_{ji} \mid A_j \rightarrow A_i \in E(\mathcal{B})\}$ for every $0 \leq i \leq n$. As \mathcal{B} has $O(n)$ edges, a total of $O(n^2)$ arithmetic operations are performed. By (sep₁)–(sep₄), $s_i^k \in \mathbb{N}$ when u is integer, thus $O(1)$ time is required by each operation. However, as it was noted by Klavík et al. [18], we cannot assume $O(1)$ time per operation when u is non-integer. The inconvenient is that to compare two fractional values a_1/b_1 and a_2/b_2 we have to multiply them with a common multiple of b_1 and b_2 . Thus, a priori, the number of bits used to represent s_j^{k-1} could be large, and the operations required to compute s_i^k could take more than constant time.

It turns out that we can represent s_i with $O(1)$ bits in a simple manner. The idea is to use a *distance tuple* $t = \langle b, [c], [\ell], [d], [d_s] \rangle$ with $b \in \{d_\ell(A), d_r(A) \mid A \in \mathcal{A}\} \cup \{-\infty\}$, $[c], [\ell] \in \mathbb{Z}$, and $[d], [d_s] \in \mathbb{N}$, in order to represent the rational

$$[t] = b + [c]c + [\ell]\ell + [d]d + [d_s]d_s.$$

So, for instance, we can represent $s_{ij} = \text{sep}(A_i \rightarrow A_j)$ as in the following table, where $q \in \{0, 1\}$ equals 0 if and only if $A_i \rightarrow A_j$ is internal.

Type	b	$[c]$	$[\ell]$	$[d]$	$[d_s]$
Step	0	$-q$	0	1	1
Nose	0	$-q$	1	1	0
Hollow	0	q	-1	1	0
Bound ($i = 0$)	$d_\ell(A_i)$	0	0	0	0
Bound ($j = 0$)	$d_r(A_i)$	-1	0	0	0

Analogously, we implement $s_i = \mathbf{d}^*\text{sep}(A_0, A_i)$ using a distance tuple for every $1 \leq i \leq n$. Just note that if s_0 is ever updated, then \mathcal{B} has a cycle of positive weight, thus we can immediately halt Bellman-Ford in failure. By doing so we observe, by invariant, that $s_i^k = [t]$ for every iteration k and some distance tuple $t = \langle [c], [\ell], [d], [d_s], b \rangle$ in which $-n \leq [c], [\ell] \leq n$, $0 \leq [d], [d_s] \leq n$.

With the above implementation, each arithmetic operation performed by Bellman-Ford costs $O(1)$ time, as it involves only $O(1)$ of the input values. We conclude, therefore, that u -REP can be solved in $O(n^2)$ time, even when u is non-integer. As far as our knowledge extends, this is the first polynomial algorithm to solve this problem.

By definition, (INT)BOUNDREP is solvable if and only if u -REP is solvable for some (integer) UCA descriptor u . The main difference between both problems is that d is an input of u -REP whereas a feasible d must be found by BOUNDREP. A simple solution for (INT)BOUNDREP is to invoke the above algorithm with a small enough value of d . For instance, if $a_1/b_1, \dots, a_k/b_k$ are the bounds of d_ℓ and d_r , then we can take $d = \frac{1}{B}$ where $B = \prod_{i=1}^k b_i$. In other words, we transform every weight of sep into an integer before invoking Bellman-Ford in the algorithm above. This algorithm is efficient when d consumes $O(1)$ bits, e.g., when u is integer. But, it is not efficient in the general case as to compare two distance tuples we need to operate with d .

An alternative solution for BOUNDREP is to find any $d < d_*$, where d_* is the maximum such that \mathcal{M} is equivalent to a $(c, \ell, d_*, d_s, d_\ell, d_r)$ -CA model. To find d we invoke the algorithm for u -REP, but instead giving an input number for d , we just think of d as a placeholder for a value lower than or equal to d_* . As before, $s_{ij} = \text{sep}(A_i \rightarrow A_j)$ and $s_i = \mathbf{d}^*\text{sep}(A_0, A_i)$ are encoded with distance tuples $[t_{ij}]$ and $[t_i]$, respectively. However, we re-implement the comparison operator to cope with the fact that d is an indeterminate value.

Let $[d_i]$ and $[d_{ij}]$ be the coefficients of t_{ij} and t_i that multiply d , respectively. For a distance tuple $t = \langle b, [c], [\ell], [d], [d_s] \rangle$, let

$$\|t\| = [t] - [d]d = b + [c]c + [\ell]\ell + [d_s]d_s.$$

The main observation is that $\mathbf{d}^*\text{sep}(A_0, A_i) < \mathbf{d}^*\text{sep}(A_0, A_j) + \text{sep}(A_j \rightarrow A_i)$ if and only if

- $\|s_i\| < \|s_{ji}\| + \|s_j\|$, or
- $\|s_i\| = \|s_{ji}\| + \|s_j\|$ and $[d_i] < [d_j] + [d_{ji}]$.

Then, every arithmetic operation costs $O(1)$ time as it involves only $O(1)$ input values.

If Bellman-Ford ends in success, then $d = \min\{\mathbf{d}^*\text{sep}(A_0, A_j) - \mathbf{d}^*\text{sep}(A_0, A_i) \mid A_i \rightarrow A_j \in E(\mathcal{B})\} = \min\{[t_i] - [t_j] \mid A_i \rightarrow A_j \in E(\mathcal{B})\} > 0$ can be obtained in $O(n)$ time. By Theorem 1, the algorithm is correct as $\text{sep}(\mathcal{W}) \leq 0$ for every cycle \mathcal{W} of \mathcal{B} . As for the certification problem, note that d consumes $O(1)$ bits, thus we can output $\mathcal{U}(\mathcal{M})$ in $O(n)$ time. Moreover, any cycle of positive weight found by Bellman-Ford can be used for negative certification.

Theorem 2. BOUNDREP, INTBOUNDREP, and u -REP can be solved in $O(n^2)$ time and $O(n)$ space.

3.2 The separation of a boundless walk

The cycles of \mathcal{B} with maximum sep-values play a fundamental role when deciding if \mathcal{M} admits an equivalent u -CA model, as shown in Theorem 1. The purpose of this section is to analyze how do these separations look like in the boundless synthetic graph. The (*boundless*) *synthetic graph* of \mathcal{M} is just $\mathcal{S}(\mathcal{M}) = \mathcal{B}(\mathcal{M}) \setminus A_0$; for the sake of simplicity, we drop the parameter \mathcal{M} as usual. The main tool that we apply is a pictorial description of \mathcal{S} , that generalizes the work of Mitas [27] on PIG models (see Section 8.1). Roughly speaking, Mitas arranges the vertices of \mathcal{S} into a matrix, where the row and column of A_i correspond to its height (cf. below) and the number of internal hollows of some paths from A_1 , respectively.

Let $\mathcal{M} = (C, \mathcal{A})$ be a PCA model with arcs $A_1 < \dots < A_n$. The *height* $h(A_i)$ of A_i ($1 \leq i \leq n$) is recursively defined as follows:

$$h(A_i) = \begin{cases} 0 & \text{if } s(A_i) < t(A_1) \\ 1 + h(A_j) & \text{otherwise, where } A_j \text{ is the maximum such that } t(A_j) < s(A_i). \end{cases}$$

The *height* of \mathcal{M} is defined as $h(\mathcal{M}) = h(A_n)$; note that $h(\mathcal{M}) \geq 1$ (because \mathcal{M} is not trivial). For the sake of notation, we drop the parameter \mathcal{M} as usual. In Figure 1, the vertices are drawn in levels according to their height.

It is important to note that internal noses and steps “jump” to higher or equal levels, while external noses and steps “jump” to lower levels. Similarly, internal hollows “jump” to equal or lower levels while external hollows “jump” to higher levels. We need a more explicit description of how does the height change when an edge is traversed. In general, we say that $A_i \rightarrow A_j$ has δ *jump* for $\delta = h(A_j) - h(A_i)$. For the sake of notation, we refer to noses (resp. steps, hollows) with δ jump simply as δ -noses (resp. δ -steps, δ -hollows).

It is not hard to see (check Figure 1) that \mathcal{S} has three kinds of noses, namely 1-, $(-h)$ -, and $(1-h)$ -noses. Moreover, if $A_i \rightarrow A_j$ is either a $(-h)$ - or $(1-h)$ -nose, then $h(A_j) = 0$. Similarly, there are three kinds of steps, namely 0-, 1-, and $(-h)$ -steps, and four kinds of hollows, namely 0-, (-1) -, h -, and external $(h-1)$ -hollows. Note that we need to differentiate between internal 0-hollows and external $(h-1)$ -hollows when $h = 1$. For the sake of simplicity, we will refer to $A_i \rightarrow A_j$ as an $(h-1)$ -hollow to mean that $A_i \rightarrow A_j$ is an **external** $(h-1)$ -hollow. We emphasize that no confusions are possible because \mathcal{M} has no external 0-hollows; otherwise A_i and A_j would cover the circle of \mathcal{M} . Observe that, as it happens with noses, $h(A_i) = 0$ for every h - or $(h-1)$ -hollow $A_i \rightarrow A_j$, while $h(A_1) = 0$ for the unique $(-h)$ -step $A_n \rightarrow A_1$. Obviously, the jump of a walk \mathcal{W} depends exclusively on the number of different kinds of noses, hollows and steps that it contains. We write $\nu_\delta(\mathcal{W})$, $\eta_\delta(\mathcal{W})$, and $\sigma_\delta(\mathcal{W})$ to indicate the number of δ -noses, δ -hollows, and δ -steps of \mathcal{W} , respectively. As usual, we do not write the parameter \mathcal{W} when it is clear from context. The following observation describes the jump of \mathcal{W} .

Observation 3. *If \mathcal{W} is a walk from A_i to A_j in \mathcal{S} , then*

$$h(A_j) - h(A_i) = \nu_1 + \sigma_1 - \eta_{-1} + h(\eta_h - \nu_{-h} - \sigma_{-h}) + (h-1)(\eta_{h-1} - \nu_{1-h}) \quad (1)$$

We now define two kinds of walks that are of particular interest for us. These walks correspond to what Tucker calls by the names of (a, b) -*independent* and (x, y) -*circuits* of a PCA model (see [35] and Section 7). We say that a walk of \mathcal{S} is a *nose walk* when it contains no hollows, while it is a *hollow walk* when it contains no noses and $\eta_h + \eta_{h-1} \geq \sigma_{-h}$. Note that a walk is both a nose and a hollow walk only if all its edges are steps; in general, walks that contain only steps are referred to as *step walks*. Nose and hollow walks are important because they impose lower and upper bounds for the circumference of the circle in a UCA model.

By Theorem 1, if \mathcal{M} is equivalent to a u -CA model, then $\text{sep}(\mathcal{W}_N) \leq 0$ for every nose cycle \mathcal{W}_N . By definition,

$$\text{sep}(\mathcal{W}_N) = \nu_1(\ell + d) + (\nu_{-h} + \nu_{1-h})(\ell + d - c) + (d + d_s)(\sigma_0 + \sigma_1) + (d + d_s - c)\sigma_{-h}$$

while by (1)

$$\nu_1 = -\sigma_1 + h(\nu_{-h} + \sigma_{-h}) + (h - 1)\nu_{1-h}$$

thus

$$c \geq (\ell + d) \left(h + \frac{\nu_{-h} - \sigma_1}{\nu_{1-h} + \nu_{-h} + \sigma_{-h}} \right). \quad (2)$$

For any nose walk \mathcal{W}_N , the value $r(\mathcal{W}_N) = \frac{\nu_{-h} - \sigma_1}{\nu_{1-h} + \nu_{-h} + \sigma_{-h}}$ is referred to as the *ratio* of \mathcal{W}_N , while the *nose ratio* of \mathcal{M} is $r(\mathcal{M}) = \max\{r(\mathcal{W}_N) \mid \mathcal{W}_N \text{ is a nose cycle of } \mathcal{M}\}$.

A similar analysis is enough to conclude (assuming $x/0 = \infty$) that

$$c \leq (\ell - d) \left(h + \frac{\eta_0 + \eta_h + \sigma_1}{\eta_h + \eta_{h-1} - \sigma_{-h}} \right) \quad (3)$$

for any hollow cycle \mathcal{W}_H (this is the reason why hollow cycles are restricted to $\eta_h + \eta_{h-1} \geq \sigma_{-h}$ by definition). This time, for any hollow walk \mathcal{W}_H , the value $R(\mathcal{W}_H) = \frac{\eta_0 + \eta_h + \sigma_1}{\eta_h + \eta_{h-1} - \sigma_{-h}}$ is said to be the *ratio* of \mathcal{W}_H , while $R(\mathcal{M}) = \min\{R(\mathcal{W}_H) \mid \mathcal{W}_H \text{ is a hollow cycle of } \mathcal{M}\}$ is the *hollow ratio* of \mathcal{M} . The following observation sums up equations (2) and (3); note that, as usual, we omit the parameter \mathcal{M} from r and R .

Observation 4. *For every u -CA model,*

$$(\ell + d)(h + r) \leq c \leq (\ell - d)(h + R) \quad (4)$$

By (4), \mathcal{M} is equivalent to a u -CA model only if $c = (\ell + d)(h + r) + e$ for some $e \geq 0$. The factor $(\ell + d)(h + r)$ is required for each nose cycle to fit in the model when considered in isolation, while the *extra space* e serves to accommodate the interactions between all the arcs. Note that, in general, \mathcal{M} needs not be equivalent to a u -CA model. This is not important, though, as we can always write c as $(\ell + d)(h + r) + e$; just observe that e could be negative in some cases.

We find it convenient to express $\text{sep}(\mathcal{W})$ as a function of ℓ and e , for every walk \mathcal{W} of \mathcal{S} . With this in mind, observe that the value of $\text{sep}(\mathcal{W})$ for a walk \mathcal{W} is, by definition

$$\begin{aligned} & \nu_1(\ell + d) + (\nu_{-h} + \nu_{1-h})(\ell + d - c) + \\ & (\eta_0 + \eta_{-1})(d - \ell) + (\eta_h + \eta_{h-1})(c + d - \ell) + \\ & (d + d_s)(\sigma_0 + \sigma_1) + (d + d_s - c)\sigma_{-h}. \end{aligned}$$

Applying Equation (1) and some algebraic manipulation, we conclude that

$$\text{sep}(\mathcal{W}) = (\ell + d)(h(A_j) - h(A_i) + \text{len}(\mathcal{W})) + e\text{ext}(\mathcal{W}) + \text{const}(\mathcal{W}) \quad (5)$$

where:

$$\text{len}(\mathcal{W}) = \nu_{-h} - \eta_h - \sigma_1 - \eta_0 + \text{rext}(\mathcal{W}) \quad (6)$$

$$\text{ext}(\mathcal{W}) = \eta_h + \eta_{h-1} - \nu_{-h} - \nu_{1-h} - \sigma_{-h} \quad (7)$$

$$\text{const}(\mathcal{W}, d, d_s) = 2d(\eta_{-1} + \eta_0 + \eta_h + \eta_{h-1}) + (d + d_s)(\sigma_1 + \sigma_0 + \sigma_{-h}) \quad (8)$$

The values $\text{len}(\mathcal{W})$, $\text{ext}(\mathcal{W})$, and $\text{const}(\mathcal{W}, d, d_s)$ are the *length*, *extra*, and *constant factors* of \mathcal{W} , and \mathcal{W} , d , and d_s are omitted as usual. We emphasize that the triplet $(\text{len}, \text{ext}, \text{const})$ is,

in some sort, a generalization to what Mitas takes as the column in his pictorial representation (see [27] and Section 8.1). The main difference is that the external edges can be disregarded from \mathcal{S} when \mathcal{M} is a PIG model. Mitas also discards the 0-hollows and the steps to define the column, thus $(\text{len}, \text{ext}, \text{const})$ gets reduced to $(0, 0, 2\eta_{-1})$.

There are at least two advantages of expressing sep as a polynomial with indeterminates ℓ and e and coefficients len , ext and const . First, by Theorem 1, we can see at first sight that \mathcal{M} is equivalent to a (c, ℓ) -CA model whenever either $\text{len}(\mathcal{W}) < 0$ or $\text{len}(\mathcal{W}) = 0$ and $\text{ext}(\mathcal{W}) < 0$ for every cycle \mathcal{W} . Just take large enough values for ℓ and e . In particular, observe that $\text{len}(\mathcal{W}) < 0$ for every cycle when \mathcal{M} is a PIG model; thus, this is just one more proof of the fact that every PIG model is equivalent to an UIG model. The second advantage is that, obviously, the factors depend only on the structure of \mathcal{S} and not on the weighing function sep . In fact, we can compute $\text{ext}(\mathcal{W})$ by means of the edge weighing ext (the overloaded notation is intentional) of \mathcal{S} such that

$$\text{ext}(A_i \rightarrow A_j) = \begin{cases} 1 & \text{if } A_i \rightarrow A_j \text{ is an } h\text{-hollow or } (h-1)\text{-hollow} \\ -1 & \text{if } A_i \rightarrow A_j \text{ is a } (-h)\text{-nose, a } (1-h)\text{-nose, or a } (-h)\text{-step} \\ 0 & \text{otherwise} \end{cases}$$

We can compute $\text{len}(\mathcal{W})$ and $\text{const}(\mathcal{W}, d, d_s)$ in a similar fashion with the corresponding edge weighings len and const_{d, d_s} .

4 Efficient Tucker's characterization

In this section we give an alternative proof of Tucker's characterization, taking advantage of the framework of synthetic graphs. In short, Tucker's theorem states that \mathcal{M} is equivalent to some UCA model if and only if $a/b < x/y$ for every (a, b) -independent and every (x, y) -circuit of \mathcal{M} [35]. As already mentioned (and proven in Section 7) the nose and hollow cycles of \mathcal{S} are the equivalents of the (a, b) -independents and (x, y) -circuits of \mathcal{M} . Moreover, the maximal and minimal values of a/b and x/y are somehow related to r and R , respectively. Thus, intuition tells us that we should be able to prove that \mathcal{M} is equivalent to a UCA model if and only if $r < R$. This is equivalence (i) \Leftrightarrow (ii) of Theorem 6 below.

Though equivalence (i) \Leftrightarrow (ii) is not new, our proof of this fact is new and somehow simple. One of the main features about Theorem 6 is that it exhibits new characterizations that can be used for positive and negative certification. In particular, it shows how to obtain an integer (c, ℓ) -CA model equivalent to \mathcal{M} with c and ℓ polynomial in n . The existence of such models was questioned by Durán et al. in [8] and proved by Lin and Szwarcfiter in [24] by means of feasible circulations.

Before stating Theorem 6, we study the relation between sep and the ratios of \mathcal{M} . Recall that the sep -values of nose and hollow cycles impose the lower and upper bounds described by (4), respectively. The reason to consider only nose and hollow cycles is that they have the largest sep -values when c and ℓ are large, as it follows from (5) and the next lemma.

Lemma 5. *For any walk \mathcal{W} of \mathcal{S} there exists either a nose or hollow walk \mathcal{W}' of \mathcal{S} starting and ending at the same vertices as \mathcal{W} such that $\text{len}(\mathcal{W}) \leq \text{len}(\mathcal{W}')$ and $\text{ext}(\mathcal{W}) \leq \text{ext}(\mathcal{W}')$.*

Proof. The proof is by induction on $\nu(\mathcal{W}) \times \eta(\mathcal{W})$ and $\sigma(\mathcal{W}) + 1$, the base case of which is trivial. Suppose, then, that \mathcal{W} has at least one nose and one hollow. So, \mathcal{W} must have a subwalk $\mathcal{W}_{1i} = B_1, \dots, B_i$ such that $B_1 \rightarrow B_2$ is a nose, B_2, \dots, B_{i-1} is a step walk, and $B_{i-1} \rightarrow B_i$ is a hollow. Observe that $i \geq 3$, because $t(B_1)s(B_2)$ are consecutive and thus $B_2 \rightarrow B_1$ is not a hollow.

Consider first the case in which \mathcal{W}_{1i} is not a path, thus it contains a cycle $\mathcal{W}_{jk} = B_j, \dots, B_k = B_j$ ($j < k$) that must have at least one 1-step or 0-hollow. Note that $\mathcal{W}_{jk} \neq \mathcal{W}_{1i}$ because otherwise B_2, \dots, B_{i-1} would pass through $B_1 = B_i$ contradicting the fact that \mathcal{W}_{jk} is a cycle. Hence, \mathcal{W}_{jk} does not contain both a nose and a hollow and $\text{len}(\mathcal{W}_{jk}) \leq 0$; recall (6). Moreover, if \mathcal{W}_{jk} has an external hollow (which must be $B_{i-1} \rightarrow B_i$), then it must contain the unique external step of \mathcal{S} . Therefore, $\text{ext}(\mathcal{W}_{ji}) \leq 0$ by (7), and the proof follows by induction on $\mathcal{W} \setminus \mathcal{W}_{ji}$.

Consider now the case in which \mathcal{W}_{1i} is a path and let \mathcal{W}'_{1i} be the step path from B_1 to B_i . We claim that $\text{len}(\mathcal{W}_{1i}) = \text{len}(\mathcal{W}'_{1i})$ and $\text{ext}(\mathcal{W}_{1i}) = \text{ext}(\mathcal{W}'_{1i})$, in which case the proof follows by induction on $(\mathcal{W} \setminus \mathcal{W}_{1i}) \cup \mathcal{W}'_{1i}$. Since \mathcal{W}_{1i} is a path, it follows that either $B_2 > B_i$ or $B_i > B_1$, which leaves us with only five possible combinations for the heights of B_1, B_2, B_{i-1} , and B_i , all of which are analyzed in the table below. The claim is therefore true.

$h(B_1)$	$h(B_2)$	$h(B_{i-1})$	$h(B_i)$	$\text{len}(\mathcal{W}_{1i})$	$\text{len}(\mathcal{W}'_{1i})$	$\text{ext}(\mathcal{W}_{1i})$	$\text{ext}(\mathcal{W}'_{1i})$
x	y	y	x	0	0	0	0
h	0	0 or 1	0	$-r$	$-r$	-1	-1
$h-1$	0	0 or 1	0	$-1-r$	$-1-r$	-1	-1
$h-1$	h or 0	0	h	-1	-1	0	0
$h-1$	h	0	$h-1$	0	0	0	0

□

The above lemma brings us closer to Tucker's characterization, as it shows that any cycle with $\text{sep} > 0$ can be transformed into a hollow or nose cycle; thus, the existence of a UCA model equivalent to \mathcal{M} is reduced to how its ratios look like. One of the salient features of our proof is that it builds an efficient UCA model \mathcal{U} equivalent to \mathcal{M} . The idea is to take $\mathcal{U} = \mathcal{U}(\mathcal{M}, c, \ell)$ as in Theorem 1 for some appropriate values of c and ℓ . Observe that $d_\ell = d_r = 0$ in this case, hence we can replace \mathcal{B} and A_0 with \mathcal{S} and A_1 in the definition of \mathcal{U} . In principle, $O(n^2)$ time is required to compute all the values of $\text{dsep}(A_1, A_i)$, because \mathcal{S} is not an acyclic graph. By taking some appropriate values for c (or e) and ℓ , we can remove all these cycles so as to reduce the time complexity to $O(n)$. With this in mind, we say that an edge $A_i \rightarrow A_j$ of \mathcal{S} is *redundant* when either

(red₁) $\text{dlen}(A_1, A_j) > \text{dlen}(A_1, A_i) + \text{len}(A_i \rightarrow A_j)$, or

(red₂) $\text{dlen}(A_1, A_j) = \text{dlen}(A_1, A_i) + \text{len}(A_i \rightarrow A_j)$ and
 $(\text{dext} \circ \text{dlen})(A_1, A_j) > (\text{dext} \circ \text{dlen})(A_1, A_i) + \text{len}(A_i \rightarrow A_j)$.

Roughly speaking, $A_i \rightarrow A_j$ is redundant when it plays no role on the separation between $s(A_1) = 0$ and $s(A_j)$ for large values of ℓ and not-so-large values of e . (Recall that $(\text{dext} \circ \text{dlen})$ is the ext-distance restricted only to those paths with maximum len-distance.) The reduction of $\mathcal{S}(\mathcal{M})$ is the digraph $\mathcal{R}(\mathcal{M})$ obtained after removing all the redundant edges of $\mathcal{S}(\mathcal{M})$; as usual, we omit the parameter \mathcal{M} . Theorem 6 includes Tucker's characterization as equivalence (i) \Leftrightarrow (ii).

Theorem 6. *Let \mathcal{M} be a PCA model with arcs $A_1 < \dots < A_n$. Then, the following statements are equivalent:*

- (i) \mathcal{M} is equivalent to a UCA model.
- (ii) $r < R$.
- (iii) $\text{len}(\mathcal{W}) < 0$ for every hollow cycle \mathcal{W} of \mathcal{S} .

(iv) either $\text{len}(\mathcal{W}) < 0$ or $\text{len}(\mathcal{W}) = 0$ and $\text{ext}(\mathcal{W}) < 0$, for every cycle \mathcal{W} of \mathcal{S} .

(v) \mathcal{R} is acyclic.

(vi) $\mathbf{d}^* \text{sep}_{(c,\ell)}(\mathcal{S}, A_1, A_i) = \mathbf{d} \text{sep}_{(c,\ell)}(\mathcal{R}, A_1, A_i)$ for every $1 \leq i \leq n$, where $c = (\ell+1)(h+r)+e$, $(\ell+1) = r_2 e^2$, $e = 4n$ and $r = r_1/r_2$ for $r_1, r_2 \in \mathbb{N}$.

(vii) $\mathcal{U}(\mathcal{M}, c, \ell)$ is an integer (c, ℓ) -CA model equivalent to \mathcal{M} for c and ℓ as in (vi).

Proof. (i) \Rightarrow (ii). This is direct consequence of (4).

(ii) \Rightarrow (iii). If \mathcal{W}_H is a hollow cycle with a nonnegative length factor, then

$$0 \leq -\sigma_1 - \eta_0 - \eta_h + r(\eta_h + \eta_{h-1} - \sigma_{-h})$$

implying (recall (3) observing that $\eta_h + \eta_{h-1} \geq \sigma_{-h}$)

$$r \geq \frac{\eta_h + \eta_0 + \sigma_1}{\eta_h + \eta_{h-1} - \sigma_{-h}} = R(\mathcal{W}_H) \geq R.$$

(iii) \Rightarrow (iv). Suppose either $\text{len}(\mathcal{W}) > 0$ or $\text{len}(\mathcal{W}) = 0$ and $\text{ext}(\mathcal{W}) \geq 0$ for some cycle \mathcal{W} of \mathcal{S} . If $\text{ext}(\mathcal{W}) \geq 0$, then the statement follows as there is a hollow cycle with a nonnegative length factor by Lemma 5. Otherwise, there is a nose cycle \mathcal{W}_N with positive length factor by Lemma 5, so

$$0 < \nu_{-h} - \sigma_1 - r(\nu_{-h} + \nu_{1-h} + \sigma_r)$$

implying (recall (2))

$$r < \frac{\nu_{-h} - \sigma_1}{\nu_{-h} + \nu_{1-h} + \sigma_{-h}} = r(\mathcal{W}_N) \leq r,$$

which is impossible.

(iv) \Rightarrow (v). Suppose \mathcal{R} has some cycle $\mathcal{W} = B_1, \dots, B_k$ with $B_1 = B_k$. By (red₁),

$$\mathbf{d} \text{len}(A_1, B_{i+1}) \leq \mathbf{d} \text{len}(A_1, B_i) + \text{len}(B_i \rightarrow B_{i+1}). \quad (\text{a})$$

Then, by induction,

$$\mathbf{d} \text{len}(A_1, B_1) = \mathbf{d} \text{len}(A_1, B_k) \leq \mathbf{d} \text{len}(A_1, B_1) + \text{len}(\mathcal{W}),$$

which implies that $\text{len}(\mathcal{W}) \geq 0$. Moreover, $\text{len}(\mathcal{W}) = 0$ only if (a) holds by equality for every $1 \leq i \leq k$, thus

$$(\mathbf{d} \text{ext} \circ \mathbf{d} \text{len})(A_1, B_{i+1}) \leq (\mathbf{d} \text{ext} \circ \mathbf{d} \text{len})(A_1, B_i) + \text{ext}(B_i \rightarrow B_{i+1})$$

by (red₂), implying $\text{ext}(\mathcal{W}) \geq 0$ by induction.

(v) \Rightarrow (vi). Taking into account that \mathcal{R} is acyclic and every walk of \mathcal{R} is also a walk of \mathcal{S} , it follows that $\mathbf{d}^* \text{sep}(\mathcal{S}, A_1, A_i) \geq \mathbf{d} \text{sep}(\mathcal{R}, A_1, A_i)$ for every $1 \leq i \leq n$.

For the remaining inequality suppose, by induction, that $\text{sep}(\mathcal{W}) \leq \mathbf{d} \text{sep}(\mathcal{R}, A_1, A_i)$ for every walk \mathcal{W} of \mathcal{S} that goes from A_1 to A_i whose length is at most $k-1 < n$. Consider any walk $\mathcal{W} = B_1, \dots, B_{k+1}$ of \mathcal{S} from $A_1 = B_1$, and let

- $\mathcal{W}_j^{\mathcal{R}}$ be a walk of \mathcal{R} from B_1 to B_j with $\text{sep}(\mathcal{W}_j^{\mathcal{R}}) = \mathbf{d} \text{sep}(\mathcal{R}, B_1, B_j)$, and
- $\mathcal{W}_{k+1}^{\mathcal{S}}$ be the walk obtained by traversing $B_k \rightarrow B_{k+1}$ after $\mathcal{W}_k^{\mathcal{R}}$.

By inductive hypothesis, $\text{sep}(\mathcal{W}) \leq \text{sep}(\mathcal{W}_k^{\mathcal{R}}) + \text{sep}(B_k \rightarrow B_{k+1}) = \text{sep}(\mathcal{W}_{k+1}^{\mathcal{S}})$, thus $\text{sep}(\mathcal{W}) \leq \text{sep}(\mathcal{W}_{k+1}^{\mathcal{R}})$ when $B_k \rightarrow B_{k+1}$ is an edge of \mathcal{R} . Suppose, then, that $B_k \rightarrow B_{k+1}$ is redundant in \mathcal{S} , and consider the two possibilities according to (red₁) and (red₂).

Case 1: (red₁) is true. Note that since no edge of $\mathcal{W}_{k+1}^{\mathcal{R}}$ is redundant, it follows by induction that $\text{len}(\mathcal{W}_j^{\mathcal{R}}) = \mathbf{dlen}(A_1, B_j)$ for every $1 \leq j \leq k+1$. Hence,

$$\text{len}(\mathcal{W}_{k+1}^{\mathcal{R}}) = \mathbf{dlen}(A_1, B_{k+1}) > \mathbf{dlen}(A_1, B_k) + \text{len}(B_k \rightarrow B_{k+1}) = \text{len}(\mathcal{W}_{k+1}^{\mathcal{S}}).$$

Now, taking into account that every term of the length factor is a multiple of either 1 or $r = r_1/r_2$ in (6), we obtain that

$$(\ell + 1)(\text{len}(\mathcal{W}_{k+1}^{\mathcal{R}}) - \text{len}(\mathcal{W}_{k+1}^{\mathcal{S}})) = r_2 e^2 (\text{len}(\mathcal{W}_{k+1}^{\mathcal{R}}) - \text{len}(\mathcal{W}_{k+1}^{\mathcal{S}})) \geq e^2.$$

By (5), we obtain that

$$\begin{aligned} \text{sep}(\mathcal{W}_{k+1}^{\mathcal{R}}) &\geq (\ell + 1)(h(B_{k+1}) + \text{len}(\mathcal{W}_{k+1}^{\mathcal{R}})) - en \\ \text{sep}(\mathcal{W}_{k+1}^{\mathcal{S}}) &\leq (\ell + 1)(h(B_{k+1}) + \text{len}(\mathcal{W}_{k+1}^{\mathcal{S}})) + ek + 2k, \end{aligned}$$

thus,

$$\text{sep}(\mathcal{W}_{k+1}^{\mathcal{R}}) - \text{sep}(\mathcal{W}_{k+1}^{\mathcal{S}}) \geq e^2 - e(k + n) - 2k \geq 8n - 2k \geq 2.$$

Case 2: (red₁) is false, thus (red₂) holds. As before, we observe by induction that $\text{len}(\mathcal{W}_j^{\mathcal{R}}) = \mathbf{dlen}(A_1, B_j)$ and, thus, $\text{ext}(\mathcal{W}_j^{\mathcal{R}}) = (\mathbf{dext} \circ \mathbf{dlen})(A_1, B_j)$. Consequently, by (red₂),

$$\text{ext}(\mathcal{W}_{k+1}^{\mathcal{R}}) > (\mathbf{dext} \circ \mathbf{dlen})(A_1, B_k) + \text{ext}(B_k \rightarrow B_{k+1}) = \text{ext}(\mathcal{W}_{k+1}^{\mathcal{S}}).$$

Since (red₁) is true, it follows that $\text{len}(\mathcal{W}_{k+1}^{\mathcal{S}}) = \mathbf{dlen}(A_1, B_{k+1})$. Then, by (5),

$$\text{sep}(\mathcal{W}_{k+1}^{\mathcal{R}}) - \text{sep}(\mathcal{W}_{k+1}^{\mathcal{S}}) \geq e(\text{ext}(\mathcal{W}_{k+1}^{\mathcal{R}}) - \text{ext}(\mathcal{W}_{k+1}^{\mathcal{S}})) - \text{const}(\mathcal{W}_{k+1}^{\mathcal{S}}) \geq 4n - 2k \geq 2.$$

We conclude, therefore, that $\text{sep}(\mathcal{W}) \leq \mathbf{dsep}(\mathcal{R}, A_1, A_i)$ for every walk \mathcal{W} of \mathcal{S} that goes from A_1 to A_i whose length is at most n . By induction, this implies that $\text{sep}(\mathcal{W}) \leq \mathbf{dsep}(\mathcal{R}, A_1, A_i)$ whatever the length of \mathcal{W} is, thus $\mathbf{d}^*\text{sep}(\mathcal{S}, A_1, A_i) = \mathbf{dsep}(\mathcal{R}, A_1, A_i)$.

(vi) \Rightarrow (vii) Since $\mathbf{d}^*\text{sep}(\mathcal{S}, A_1, A_i) = \mathbf{dsep}(\mathcal{R}, A_1, A_i)$ for every $1 \leq i \leq n$ and \mathcal{S} is strongly connected, it follows that $\mathbf{d}^*\text{sep}(\mathcal{S}, A_i, A_j) = \mathbf{dsep}(\mathcal{S}, A_i, A_j)$ for every $1 \leq i, j \leq n$. Hence $\text{sep}(\mathcal{W}) \leq 0$ for every cycle \mathcal{W} of \mathcal{S} and the implication follows by Theorem 1 (note that c and ℓ are integer values).

(vii) \Rightarrow (i). Trivial. \square

Theorem 6 has some nice algorithmic consequences REP when combined with Theorem 1. For any input PCA model \mathcal{M} we solve u -REP for the UCA descriptor u implied by statement (vi). As a byproduct, we either obtain a UCA model equivalent to \mathcal{M} or a cycle of \mathcal{S} that can be used for negative certification. The algorithm costs $O(n^2)$ time, plus the time and space required so as to compute $r(\mathcal{M})$. In Section 6 we show a not-so-hard $O(n)$ time variation of this algorithm, taking advantage of the reduction of \mathcal{S} . However, we first discuss how r can be found.

5 The recognition algorithm by Kaplan and Nussbaum

Translated to synthetic graphs, Tucker's characterization (equivalence (i) \Leftrightarrow (ii) of Theorem 6) states that \mathcal{M} is equivalent to no UCA model if only if \mathcal{S} has nose and hollow cycles \mathcal{W}_N and \mathcal{W}_H such that $r(\mathcal{W}_N) \geq R(\mathcal{W}_H)$. The original proof by Tucker does not show how to obtain such cycles. More than thirty years later, in [8], Durán et al. described the first polynomial algorithm to obtain such cycles with a rather complex implementation. A few years later, in [15],

Kaplan and Nussbaum improved this algorithm so as to run in $O(n)$ time while simplifying the implementation. The purpose of this section is to translate the algorithm by Kaplan and Nussbaum in terms of the synthetic graph. The proof of correctness is simple, short, and rather intuitive, while the implementation of the algorithm is quite similar to the one given by Kaplan and Nussbaum.

The main concept of this section is that of greedy cycles. For any nose (resp. hollow) walk $\mathcal{W}_N = B_1, \dots, B_k$ of \mathcal{S} , we say that B_i is *greedy (in \mathcal{W}_N)* when either $B_i \rightarrow B_{i+1}$ is nose (resp. hollow) or no nose (resp. hollow) of \mathcal{S} starts at B_i . A nose (resp. hollow) cycle is *greedy* when all its vertices are greedy. In other words, \mathcal{W}_N is greedy when noses (resp. hollow) are preferred over steps. The main idea of Durán et al., which was somehow implicit in [35], is to observe that \mathcal{S} contains a greedy nose (resp. hollow) cycle of highest (resp. lowest) ratio. Then, they compute the unique greedy nose (resp. hollow) cycle starting at a vertex A , for every $A \in V(\mathcal{S})$, and keep the one with highest (resp. lowest) ratio. Note that each greedy nose (resp. hollow) cycle B_1, \dots, B_k is found k times, once for each starting vertex B_i . Kaplan and Nussbaum, instead, compute each greed nose (resp. hollow) cycle only once by taking only one vertex as the starting point.

The next lemma has a new proof that \mathcal{S} contains a greedy nose cycle of highest ratio.

Lemma 7 (see also [8, 15, 35]). *For any nose cycle \mathcal{W}_N of \mathcal{S} there exists a greedy nose cycle \mathcal{W}'_N of \mathcal{S} such that $r(\mathcal{W}_N) \leq r(\mathcal{W}'_N)$.*

Proof. The proof is trivial when \mathcal{W}_N is greedy. When \mathcal{W}_N is not greedy, we can transform it into a greedy nose cycle by traversing \mathcal{W}_N from any vertex while applying the following operation when a non-greedy vertex B_1 is found, until no more non-greedy vertices remain. Let $B_1 \rightarrow B$ be the nose from B_1 and $\mathcal{W} = B_1, \dots, B_i$ be the shortest subpath of \mathcal{W}_N such that either $B_i = B$ or $B_{i-1} \rightarrow B_i$ is a nose. The operation transforms \mathcal{W}_N into \mathcal{W}'_N by replacing \mathcal{W} with \mathcal{W}' , where \mathcal{W}' is the path formed by the nose $B_1 \rightarrow B$ followed by step path from B to B_i . Thus, it suffices to prove that $r(\mathcal{W}) \leq r(\mathcal{W}')$. Moreover, if we write r_1 and r_2 to denote the numerator and denominator of r as in (2), then $r_j(\mathcal{W}'_N) = r_j(\mathcal{W}_N) - r_j(\mathcal{W}) + r_j(\mathcal{W}')$ for $j \in \{1, 2\}$. So, it is enough to show that $r_1(\mathcal{W}') \geq r_1(\mathcal{W})$ and $r_2(\mathcal{W}') \leq r_2(\mathcal{W})$.

If $B = B_i$ then \mathcal{W} has at least one 1-step or $(-h)$ -step, while $\mathcal{W}' = B_1, B$. Thus $r_1(\mathcal{W}) \leq 0 \leq r_1(\mathcal{W}')$ and $r_2(\mathcal{W}') \leq r_2(\mathcal{W})$, and the lemma follows. Otherwise, if $B \notin \mathcal{W}$, then B_1, B_{i-1}, B, B_i appear in this order in \mathcal{S} when its steps are traversed from B_1 . It is not hard to see that $r_i(\mathcal{W}) = r_i(\mathcal{W}')$ when $B_1 \rightarrow B$ and $B_{i-1} \rightarrow B_i$ have equal jumps. This leaves us with only four possible combinations for the heights of B_1, B_{i-1}, B_i , and B when the jumps differ, and in all such cases the lemma is true (see the table below).

$h(B_1)$	$h(B_{i-1})$	$h(B)$	$h(B_i)$	$r_1(\mathcal{W})$	$r_1(\mathcal{W}')$	$r_2(\mathcal{W})$	$r_2(\mathcal{W}')$
$h - 2$	$h - 1$	$h - 1$	0	-1	-1	1	1
$h - 1$	$h - 1$	h	0	0	0	1	1
$h - 1$	h	h or 0	0	0	0	1	1
h	0	0	1	0	0	1	1

□

The proof that \mathcal{S} contains a greedy hollow cycle with lowest ratio is similar and we omit it as it not required by our algorithm. Moreover, an analogous proof is given in Lemma 9, while Section 7 shows the equivalence between hollow cycles and (x, y) -circuits.

Lemma 8 (see [15] and Section 7). *For any hollow cycle \mathcal{W}_H of \mathcal{S} there exists a greedy hollow cycle \mathcal{W}'_H of \mathcal{S} such that $R(\mathcal{W}_H) \geq R(\mathcal{W}'_H)$.*

The algorithm to compute a nose (resp. hollow) cycle with highest (resp. lowest) ratio follows easily from Lemma 7 (resp. Lemma 8). Just note that if an edge $A_i \rightarrow A_j$ belongs to a greedy nose (resp. hollow) cycle, then either $A_i \rightarrow A_j$ is a nose (resp. hollow), or there are no noses (resp. hollow) from A_i in \mathcal{S} . Then, \mathcal{W} is a greedy nose (resp. hollow) cycle of \mathcal{S} if and only if \mathcal{W} is a cycle of the digraph \mathcal{S}_N that is obtained by keeping only the noses (resp. hollows) of \mathcal{S} and the steps that go from vertices with no noses (resp. hollows). Since all the vertices in \mathcal{S}_N have out-degree 1, we can obtain all the greedy cycles in $O(n)$ time. Then, by Lemmas 7 and 8, r and R can be computed in $O(n)$ time. Furthermore, if $r > R$, then a nose and a hollow cycles \mathcal{W}_N and \mathcal{W}_H with $r(\mathcal{W}_N) = r$ and $R(\mathcal{W}_H) = R$ are obtained as a byproduct.

6 Efficient construction of UCA models

In [8], Durán et al. ask if there exists an integer (c, ℓ) -CA model equivalent to a PCA model \mathcal{M} such that c and ℓ are bounded by a polynomial in n . If affirmative, they also inquire whether such a model can be found in $O(n)$ time. Both questions were affirmatively answered by Lin and Szwarcfiter in [24], who showed how to reduce the problem of finding such a (c, ℓ) -CA model to a circulation problem. Their algorithm and its correctness have nothing to do with Theorem 6 and it is not easy to see how a forbidden subgraph can be obtained for certification (that is, without invoking a second recognition algorithm, as the one by Kaplan and Nussbaum). Kaplan and Nussbaum ask for a *unified* certification algorithm. We provide such an algorithm in this section.

Our algorithm is based on the equivalence (i) \Leftrightarrow (v) \Leftrightarrow (vi) of Theorem 6. That is, the algorithm just checks whether $\mathcal{R}(\mathcal{M})$ is acyclic. If affirmative, then $\mathcal{U}(c, \ell)$ can be taken as the positive certificate by Theorem 1, where c and ℓ are defined as in Theorem 6(vi). Otherwise, a nose cycle with ratio $r(\mathcal{M})$ combined with a cycle of $\mathcal{R}(\mathcal{M})$ form the negative certificate. Of course, testing if \mathcal{R} is acyclic and finding a cycle in \mathcal{R} both cost $O(n)$ time. When \mathcal{R} is acyclic, we can compute \mathbf{dsep} in $O(n)$ time using \mathcal{R} so as to build $\mathcal{U}(c, \ell)$. Hence, the difficulty of the algorithm is on finding \mathcal{R} .

By definition, \mathcal{R} is obtained by removing the redundant edges of \mathcal{S} ; this can be done in $O(n)$ time once the redundant edges of \mathcal{S} are found. In turn, recall that an edge $A_i \rightarrow A_j$ is redundant if and only if

- (red₁) $\mathbf{dlen}(A_1, A_j) > \mathbf{dlen}(A_1, A_i) + \text{len}(A_i \rightarrow A_j)$, or
- (red₂) $\mathbf{dlen}(A_1, A_j) > \mathbf{dlen}(A_1, A_i) + \text{len}(A_i \rightarrow A_j)$ and
 $(\mathbf{dlen} \circ \mathbf{dext})(A_1, A_j) > (\mathbf{dlen} \circ \mathbf{dext})(A_1, A_i) + \text{ext}(A_i \rightarrow A_j)$.

Thus, to locate the redundant edges we should find a path \mathcal{W}_i from A_1 to A_i such that $\text{len}(\mathcal{W}_i) = \mathbf{dlen}(A_1, A_i)$ and $\text{ext}(\mathcal{W}_i) = (\mathbf{dlen} \circ \mathbf{dext})(A_1, A_i)$ for every $1 \leq i \leq n$. By Lemma 5, \mathcal{W}_i is either a nose or hollow cycle. That is, $\mathcal{W}_i \in \{\mathcal{W}_{N,i}, \mathcal{W}_{H,i}\}$, where $\mathcal{W}_{N,i}$ is the nose cycle such that

- (gn₁) $\text{len}(\mathcal{W}_{N,i}) = \max\{\text{len}(\mathcal{W}) \mid \mathcal{W} \text{ is a nose path from } A_1 \text{ to } A_i\}$, and
- (gn₂) $\text{ext}(\mathcal{W}_{N,i}) = \max\{\text{ext}(\mathcal{W}) \mid \mathcal{W} \text{ is a nose path from } A_1 \text{ to } A_i \text{ with } \text{len}(\mathcal{W}) = \text{len}(\mathcal{W}_{N,i})\}$,

while $\mathcal{W}_{H,i}$ is defined analogously by replacing noses with hollows. The remainder of this section is devoted to the problems of finding $\mathcal{W}_{N,i}$ and $\mathcal{W}_{H,i}$.

Say that a nose (resp. hollow) walk $\mathcal{W} = B_1, \dots, B_k$ is *greedy* when there exists $B_i \in \mathcal{W}$ such that B_j is greedy for every $1 \leq j \leq i$, while B_i, \dots, B_k is a step walk. In other words, \mathcal{W} is greedy when noses (resp. hollows) are preferred until some point in which only steps follow.

It turns out that $\mathcal{W}_{N,i}$ and $\mathcal{W}_{H,i}$ are greedy paths. The proof of this fact is analogous to those of Lemmas 7 and 8, yet we include it for the sake of completeness.

Lemma 9. *For any nose (resp. hollow) walk \mathcal{W}_N of \mathcal{S} there exists a greedy nose (resp. hollow) walk \mathcal{W}'_N of \mathcal{S} joining the same vertices such that either $\text{len}(\mathcal{W}_N) < \text{len}(\mathcal{W}'_N)$ or $\text{len}(\mathcal{W}_N) = \text{len}(\mathcal{W}'_N)$ and $\text{ext}(\mathcal{W}_N) \leq \text{ext}(\mathcal{W}'_N)$.*

Proof. The proof is by induction on $|\mathcal{W}_N| - p$ and $|\mathcal{W}_N|$, where p is the position of the first non-greedy vertex of \mathcal{W}_N . The base case in which p is greater than the position last nose (resp. hollow) of \mathcal{W}_N is trivial. Suppose, then, that B_1 is the first non-greedy vertex of \mathcal{W}_N and that B_1 appears before the last nose (resp. hollow) of \mathcal{W}_N . Let $B_1 \rightarrow B$ be the nose (resp. hollow) from B_1 and consider the following cases.

Case 1: \mathcal{W}_N is a nose walk. Let $\mathcal{W} = B_1, \dots, B_i$ be the shortest subpath of \mathcal{W}_N such that either $B_{i-1} \rightarrow B_i$ is a nose or $B_i = B$, \mathcal{W}' be the path formed by the nose $B_1 \rightarrow B$ followed by the step path from B to B_i , and \mathcal{W}'_N be the nose walk obtained from \mathcal{W}_N by replacing \mathcal{W} by \mathcal{W}' . Clearly, the position of the first non-greedy vertex of \mathcal{W}'_N is greater than p , thus, by induction, it suffices to show that $w(\mathcal{W}') \geq w(\mathcal{W})$ for $w \in \{\text{len}, \text{ext}\}$ because $w(\mathcal{W}'_N) = w(\mathcal{W}_N) - w(\mathcal{W}) + w(\mathcal{W}')$. If $B_i = B$, then either \mathcal{W} contains at least one 1-step or $B_1 \rightarrow B_i$ is a $(-h)$ -nose, thus $\text{len}(\mathcal{W}') > \text{len}(\mathcal{W})$. Otherwise, B_1, B_{i-1}, B, B_i appear in this order in \mathcal{S} when the steps are traversed from B_1 . As in Lemma 7, it is not hard to see that $w(\mathcal{W}') = w(\mathcal{W})$ when $B_1 \rightarrow B$ and $B_{i-1} \rightarrow B_i$ have equal jumps, while, as in Lemma 7, only four cases remain otherwise. All these cases are examined in the table below.

$h(B_1)$	$h(B_{i-1})$	$h(B)$	$h(B_i)$	$\text{len}(\mathcal{W})$	$\text{len}(\mathcal{W}')$	$\text{ext}(\mathcal{W})$	$\text{ext}(\mathcal{W}')$
$h-1$	$h-1$ or h	h	0	$-r$	$-r$	-1	-1
$h-2$	$h-1$	$h-1$	0	$-1-r$	$-1-r$	-1	-1
$h-1$	h	0	0	$-r$	$-r$	-1	-1
h	0	0	1	$-r$	$-r$	-1	-1

Case 2: \mathcal{W}_N is a hollow walk. Let $\mathcal{W} = B_1, \dots, B_i$ be the shortest subpath of \mathcal{W}_N such that $B_{i-1} \rightarrow B_i$ is a hollow. If $B_i = B_j$ for some $1 \leq j < i$, then $\mathcal{W}_{ji} = B_j, \dots, B_i$ is a cycle with exactly one 1-step or 0-hollow, thus $\text{len}(\mathcal{W}_{ji}) < 0$. So, the proof follows by induction on the hollow walk $\mathcal{W}'_N = \mathcal{W}_N \setminus \mathcal{W}_{ji}$ because $\text{len}(\mathcal{W}'_N) = \text{len}(\mathcal{W}_N) - \text{len}(\mathcal{W}_{ji})$ and the position of the first non-greedy hollow is at least p . If $B_i \notin \{B_1, \dots, B_{i-1}\}$, then B, B_i, B_1, B_{i-1} appear in this order in a traversal of the steps of \mathcal{S} from B . Let \mathcal{W}' be the hollow path formed by the hollow $B_i \rightarrow B$ followed by the step path from B to B_i and observe that, as in Case 1, it suffices to prove that $w(\mathcal{W}') = w(\mathcal{W})$ for $w \in \{\text{len}, \text{ext}\}$. Moreover, both equalities hold when $B_1 \rightarrow B$ and $B_{i-1} \rightarrow B_i$ have equal jumps. The equalities hold also when either $B_1 \rightarrow B$ or $B_{i-1} \rightarrow B_i$ is a 0-hollow. Indeed, if $B_1 \rightarrow B$ is a 0-hollow, then $h(B) = h(B_i) = h(B_1)$ and $h(B_{i-1}) \neq h(B)$, while if $B_{i-1} \rightarrow B_i$ is a 0-hollow, then $h(B_i) = h(B_1) = h(B_{i-1})$ and $h(B) \neq h(B_{i-1})$. In both of these cases, $\text{len}(\mathcal{W}) = \text{len}(\mathcal{W}') = -1$ and $\text{ext}(\mathcal{W}) = \text{ext}(\mathcal{W}') = 0$. Finally, when neither $B_1 \rightarrow B$ nor $B_{i-1} \rightarrow B_i$ are 0-hollows and $B_1 \rightarrow B$ and $B_{i-1} \rightarrow B_i$ have different jumps, we are left with only six cases, as in the table below.

$h(B_1)$	$h(B_{i-1})$	$h(B)$	$h(B_i)$	$\text{len}(\mathcal{W})$	$\text{len}(\mathcal{W}')$	$\text{ext}(\mathcal{W})$	$\text{ext}(\mathcal{W}')$
0	1	h or $h-1$	0	-1	-1	0	0
0	0	$h-1$	h	$-1+r$	$-1+r$	1	1
h	0	$h-1$	h	-1	-1	0	0
h	0	$h-1$	$h-1$	0	0	0	0
$h-1$	0	$h-2$	$h-1$	-1	-1	0	0

□

Recall that our problem is to find the len and ext values for the path $\mathcal{W}_{N,i}$ satisfying (gn₁) and (gn₂), for every $1 \leq i \leq n$. We solve this problem in two phases. Clearly, there exists a unique greedy nose path \mathcal{N} beginning at A_1 that is maximal. The first phase consist of traversing $\mathcal{N} = B_1, \dots, B_k$ while $p(B_i) = (\text{len}(B_1, \dots, B_i), \text{ext}(B_1, \dots, B_i))$ is computed and stored for every $1 \leq i \leq k$. In the second phase each step $A_{i-1} \rightarrow A_i$ of \mathcal{S} is traversed while $q(A_i) = (\text{len}(\mathcal{W}_{N,i}), \text{ext}(\mathcal{W}_{N,i}))$ is computed and stored. By Lemma 9, $\mathcal{W}_{N,i}$ is a greedy nose path starting at A_1 , thus $\mathcal{W}_{N,i}$ is equal to a subpath of \mathcal{N} plus a (possibly empty) step path. Consequently, there are only two possibilities for the last edge of $\mathcal{W}_{N,i}$ according to whether $A_i \in \mathcal{N}$ or not. If $A_i \notin \mathcal{N}$, then the last edge of $\mathcal{W}_{N,i}$ must be $A_{i-1} \rightarrow A_i$, thus $q(A_i) = q(A_{i-1}) + (\text{len}(A_{i-1} \rightarrow A_i), \text{ext}(A_{i-1} \rightarrow A_i))$. Otherwise, the last edge could be $A_{i-1} \rightarrow A_i$ or the unique nose $A_j \rightarrow A_i$. In the latter case $\mathcal{W}_{N,i}$ is a subpath of \mathcal{N} . Thus, we can compute $\mathcal{W}_{N,i}$ by simply comparing the values $p(A_i)$ and $q(A_{i-1}) + (\text{len}(A_{i-1} \rightarrow A_i), \text{ext}(A_{i-1} \rightarrow A_i))$. Note that both traversals cost $O(n)$ time. The problem of finding $\mathcal{W}_{H,i}$ is analogous and it also costs $O(n)$ time.

Theorem 10. *There is a unified certified algorithm that solves REP in $O(n)$ time.*

6.1 Logspace construction of UCA models

In the full version of [21], Kobler et al. ask whether it is possible to solve REP in (deterministic) logspace. In this section we provide an affirmative answer to this question by showing that the algorithm of the previous section can be implemented so as to run in logspace. Before doing so, we briefly discuss the logspace recognition of UCA graphs, for the sake of completeness.

Kobler et al. [21] show a logspace algorithm for the recognition of PCA graphs. As a byproduct, their algorithm outputs a PCA model \mathcal{M} with arcs $A_1 < \dots < A_n$; we implement the algorithm by Kaplan and Nussbaum so as to run in logspace when \mathcal{M} is given. Let \mathcal{S}_N be the subgraph of \mathcal{S} obtained by removing all its hollows and all its steps that go from a vertex in which a nose starts. All the vertices of \mathcal{S}_N have out-degree 1. So, the nose ratio of each cycle \mathcal{W} of \mathcal{S}_N can be obtained in logspace by traversing \mathcal{W} from each of its vertices. By Lemma 7, r is the maximum among such ratios. Then, taking into account that \mathcal{S}_N can be easily computed in logspace from \mathcal{M} , we conclude that r is obtainable in logspace. An analogous algorithm can be used to compute the hollow ratio R of \mathcal{M} in logspace. By Theorem 6, \mathcal{M} is equivalent to a UCA model if and only if $r < R$. The algorithm can output, also in logspace, the nose and hollow cycles with ratios r and R , respectively. These cycles provide a negative certificate that \mathcal{M} is equivalent to no UCA models when $r \geq R$.

The logspace representation algorithm can be divided in two phases. In the first phase, \mathcal{R} is build, while, in the second phase, the UCA model is obtained from a topological sort of its vertices. Clearly, the problem of computing \mathcal{R} can be logspace reduced to querying which of the edges of \mathcal{S} are redundant. By (red₁) and (red₂), the problem of testing if $A_i \rightarrow A_j$ is redundant is logspace reduced to that of finding $\mathbf{dlen}(A_1, A_i)$ and $(\mathbf{dlen} \circ \mathbf{dext})(A_1, A_i)$. As stated in the previous section, this problem is reduced to that of computing $\text{len}(\mathcal{W}_{N,i})$, $\text{ext}(\mathcal{W}_{N,i})$, $\text{len}(\mathcal{W}_{H,i})$,

and $\text{ext}(\mathcal{W}_{H,i})$ where $\mathcal{W}_{N,i}$ is the greedy nose path from A_1 to A_i defined by (gn_1) and (gn_2) , while $\mathcal{W}_{H,i}$ is defined analogously. By Lemma 7, $\mathcal{W}_{N,i} = \mathcal{N}_{a,b}$ for some $a, b < n$, where $\mathcal{N}_{a,b}$ is the walk obtained by first traversing a edges of the unique maximal greedy path \mathcal{N} that begins at A_1 , and then traversing b steps. So, by keeping the counters a and b , we can compute the maximum among the values of $\text{len}(\mathcal{N}_{a,b})$ and $\text{ext}(\mathcal{N}_{a,b})$ for the paths $\mathcal{N}_{a,b}$ ending at A_i with $a, b < n$. Such a computation requires logspace, thus \mathcal{R} can be obtained in logspace.

Once \mathcal{R} is build, we could compute $\mathcal{U}(c, \ell)$ by finding $\text{dsep}(\mathcal{R}, A_1, A_i)$ for every $1 \leq i \leq n$. There is a major inconvenience with this approach: finding the longest path between two vertices of an acyclic digraph is a complete problem for the class of non-deterministic logspace problems. To deal with this problem we could observe that \mathcal{R} is not only an acyclic digraph but also one with a rather particular structure.

Theorem 11. *If \mathcal{M} is a PCA model, then \mathcal{S} is a toroidal digraph.*

Proof. As the theorem is implied by [27], we defer its proof to Section 8.1 (see Corollary 21). \square

Toroidal acyclic digraphs are much simpler than general acyclic digraphs. Yet, up to this date, the best algorithms to compute their longest paths run in unambiguous logspace [23]. For this reason, the UCA model computed in the second phase is a variation of $\mathcal{U}(c, \ell)$. The key idea is to observe that the reachability problem for toroidal digraphs that have a unique vertex with in-degree 0 can be solved in logspace [34]. That is, for $A_i, A_j \in V(\mathcal{R})$, the *reachability algorithm* in [34] outputs YES when there is a path from A_i to A_j in \mathcal{R} . Then, we can compute $\text{reach}(A_j) = |\{A_i \in V(\mathcal{R}) \mid \text{there is a path from } A_i \text{ to } A_j \text{ in } \mathcal{R}\}|$ in logspace for any given $A_j \in V(\mathcal{R})$. The representation algorithm takes advantage of this fact by replacing const with the easier-to-find reach . That is, the constructed UCA model $\mathcal{U}_{\text{reach}}$ is the (c, ℓ) -CA model with arcs U_1, \dots, U_n such that $c = (\ell + 1)(h + r) + e$ for $e = 4n$, $\ell + 1 = r_2 e^2$, and

$$s(U_j) = (\ell + 1)(h(A_j) + \text{dlen}(A_1, A_j)) + e(\text{dext} \circ \text{dlen})(A_1, A_j) + 2\text{reach}(A_j) \quad (9)$$

for every $1 \leq j \leq n$. Here, as in Theorem 6 (vi), $r = r_1/r_2$, thus c , ℓ , and $s(U_j)$ are integers. By the previous discussion, $\mathcal{U}_{\text{reach}}$ is obtainable in logspace. The fact that $\mathcal{U}_{\text{reach}}$ is equivalent to \mathcal{M} follows from the next theorem.

Theorem 12. *Let \mathcal{M} be a PCA model. Then, \mathcal{R} is acyclic if and only if $\mathcal{U}_{\text{reach}}$ is an integer UCA model equivalent to \mathcal{M} .*

Proof. Suppose first that \mathcal{R} is acyclic and let U_1, \dots, U_n be the arcs of $\mathcal{U}_{\text{reach}}$ as in its definition. Then, it suffices to show that:

- (a) $s(U_j) \geq s(U_i) + \ell + 1 - cq$ for every nose $U_i \rightarrow U_j$ of \mathcal{S} ,
- (b) $s(U_j) \geq s(U_i) - \ell + 1 + cq$ for every hollow $U_i \rightarrow U_j$ of \mathcal{S} , and
- (c) $s(U_j) \geq s(U_i) + 1 - cq$ for every step $U_i \rightarrow U_j$.

where $q \in \{0, 1\}$ equals 0 if and only if $A_i \rightarrow A_j$ is internal. Take any edge $A_i \rightarrow A_j$ and, for the sake of notation, let:

- $\Delta x(i, j) = x(A_j) - x(A_i)$ for $x \in \{h, \text{reach}\}$,
- $\Delta \text{len}(i, j) = \text{dlen}(A_1, A_j) - \text{dlen}(A_1, A_i) - \text{len}(A_i \rightarrow A_j)$, and
- $\Delta \text{ext}(i, j) = (\text{dext} \circ \text{dlen})(A_1, A_j) - (\text{dext} \circ \text{dlen})(A_1, A_i) - \text{ext}(A_i \rightarrow A_j)$.

By definition (9),

$$\begin{aligned}
s(U_j) &= (\ell + 1)(h(A_j) + \mathbf{dlen}(A_1, A_j)) + e(\mathbf{dext} \circ \mathbf{dlen})(A_1, A_j) + 2\text{reach}(A_j) \\
&= (\ell + 1)(h(A_i) + \Delta h(i, j) + \mathbf{dlen}(A_1, A_i) + \text{len}(A_i \rightarrow A_j) + \Delta \text{len}(i, j)) + \\
&\quad e((\mathbf{dext} \circ \mathbf{dlen})(A_1, A_i) + \text{ext}(A_i \rightarrow A_j) + \Delta \text{ext}(i, j)) + \\
&\quad 2\text{reach}(A_i) + 2\Delta \text{reach}(i, j) \\
&= s(U_i) + (\ell + 1)(\Delta h(i, j) + \text{len}(A_i \rightarrow A_j)) + e(\text{ext}(A_i \rightarrow A_j)) + \varepsilon
\end{aligned} \tag{i}$$

where $\varepsilon = (\ell + 1)\Delta \text{len}(i, j) + e\Delta \text{ext}(i, j) + 2\Delta \text{reach}(i, j)$.

Note that $\varepsilon \geq 2$. Indeed, if $A_i \rightarrow A_j$ is redundant, then either $\Delta \text{len}(i, j) > 0$ or $\Delta \text{len}(i, j) = 0$ and $\Delta \text{ext}(i, j) > 0$; thus $\varepsilon \geq 2$ as in Theorem 6 (v) \Rightarrow (vi). If $A_i \rightarrow A_j$ is not redundant, then $A_i \rightarrow A_j$ is an edge of \mathcal{R} and $\Delta \text{reach}(i, j) > 0$, while $\Delta \text{len}(i, j) = \Delta \text{ext}(i, j) = 0$. Consequently, $\varepsilon \geq 2$ regardless of whether $A_i \rightarrow A_j$ is redundant or not. Then, (a)–(c) follow by inspection, considering the 10 possible values for the jump of $A_i \rightarrow A_j$. For the sake of completeness, the table below sums up all these cases; recall that $c = h(\ell + 1) + r(\ell + 1) + e$.

Type $A_i \rightarrow A_j$	q	$\Delta h(A_i, A_j)$	$\text{len}(A_i \rightarrow A_j)$	$\text{ext}(A_i \rightarrow A_j)$	(i) $-\varepsilon$
1-nose	0	1	0	0	$s(U_i) + \ell + 1$
$(1 - h)$ -nose	1	$-h + 1$	$-r$	-1	$s(U_i) + \ell + 1 - c$
$(-h)$ -nose	1	$-h$	$1 - r$	-1	$s(U_i) + \ell + 1 - c$
(-1) -hollow	0	-1	0	0	$s(U_i) - \ell - 1$
0-hollow	0	0	-1	0	$s(U_i) - \ell - 1$
$(h - 1)$ -hollow	1	$h - 1$	r	1	$s(U_i) - \ell - 1 + c$
h -hollow	1	h	$r - 1$	1	$s(U_i) - \ell - 1 + c$
0-step	0	0	0	0	0
1-step	0	1	-1	0	0
$(-h)$ -step	1	$-h$	$-r$	-1	$s(U_i) - c$

The converse follow from Theorem 6 (i) \Rightarrow (v). □

7 (a, b) -independents and (x, y) -circuits

As stated, our algorithm outputs two cycles when \mathcal{M} is not equivalent to a UCA model: a nose cycle \mathcal{W}_N with ratio $r(\mathcal{M})$ and a cycle \mathcal{W}_H of \mathcal{R} . As in the proofs of implications (iii) \Leftrightarrow (iv) \Leftrightarrow (v), \mathcal{W}_H is a hollow cycle with a nonnegative length factor. Moreover, as in implication (ii) \Leftrightarrow (iii), $R(\mathcal{W}_H) \leq r(\mathcal{W}_N)$. Note that, in principle, this certificate needs not be equal to the one in Section 5, because \mathcal{W}_H needs not be the hollow cycle with minimum ratio. Nevertheless, this certificate is somehow analogous to the one provided by the algorithm by Kaplan and Nussbaum.

Rigorously speaking, the certificate of Section 5 is neither equal to the one given by the algorithm by Kaplan and Nussbaum. The former is a pair of nose and hollow cycles while the latter is a pair of (a, b) -independent plus (x, y) -circuit. Nose cycles can contain more vertices than the corresponding (a, b) -independents while hollow cycles can contain more vertices than the corresponding (x, y) -circuits. These added vertices are, nevertheless, redundant and can be eliminated from the certificate so as to obtain a minimal forbidden induced submodel as the negative certificate. The purpose of this section is to describe the equivalence between nose (resp. hollow) cycles and (a, b) -independents (resp. (x, y) -circuits) and how to transform one into the other and vice versa. We begin describing what are the (a, b) -independents and (x, y) -circuits.

For two arcs A_i, A_j of a PCA model \mathcal{M} , we define the *ss arc of A_i, A_j* to be the arc $(s(A_i), s(A_j))$. For a sequence of arcs $\mathcal{A} = B_1, \dots, B_k$, the *ss traversal of \mathcal{A}* is the family of arcs \mathcal{T} that contains the *ss arc* of B_i, B_{i+1} for every $1 \leq i \leq k$ (where $B_1 = B_{k+1}$). The number of *turns* of \mathcal{T} is the number of its arcs that contain the point 0 of $C(\mathcal{M})$. In simple terms, the *ss traversal* of \mathcal{A} is obtained by traversing $C(\mathcal{M})$ from $s(B_1)$ to $s(B_2)$ to \dots to $s(B_k)$ to $s(B_1)$, while its number of turns is the number of complete loops to the circle in such a traversal.

An (a, b) -independent of a PCA model \mathcal{M} is a sequence of arcs $\mathcal{A} = B_1, \dots, B_a$ such that $s(B_{i+1}) \notin B_i$ for every $1 \leq i \leq a$ and whose *ss traversal* takes b turns. Similarly, an (x, y) -circuit is a sequence of arcs B_1, \dots, B_x such that $s(B_{i+1}) \in B_i$ for every $1 \leq i \leq x$ and whose *ss traversal* takes y turns. Note that $x > 2y$ as no pair of arcs of \mathcal{M} cover the circle. An (a, b) -independent is *maximal* when a/b is maximum and a, b are relative primes, while an (x, y) -circuit is *minimal* when x/y is minimum and x, y are relative primes. As we shall shortly see, statement (i) \Leftrightarrow (ii) of Theorem 6 is equivalent to the following theorem by Tucker.

Theorem 13 ([35]). *A PCA model \mathcal{M} is equivalent to an UCA model if and only if $a/b < x/y$ for every maximal (a, b) -independent and every minimal (x, y) -circuit.*

Say that an (a, b) -independent $\mathcal{A} = B_1, \dots, B_a$ is *standard* when $s(B_i)$ is immediately preceded by an ending point in \mathcal{M} , for every $1 \leq i \leq a$. Note that if $s(B_i)$ is preceded by the beginning point of an arc A , then $B_1, \dots, B_{i-1}, A, B_{i+1}, \dots, B_a$ is also an (a, b) -independent of \mathcal{M} . Consequently, \mathcal{M} has an (a, b) -independent if and only if it has a standard (a, b) -independent.

There is a one-to-one correspondence between the standard (a, b) -independents of \mathcal{M} and the nose circuits of \mathcal{S} , as follows. Let $\mathcal{A} = B_1, \dots, B_a$ be a standard (a, b) -independent and \mathcal{W}_i be the step path of \mathcal{S} that goes from B_i to B'_i , where B'_i is the arc whose ending point immediately precedes $s(B_{i+1})$. Clearly, $B'_i \rightarrow B_{i+1}$ is a nose of \mathcal{S} , thus $\mathcal{W}(\mathcal{A}) = \mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_a$ is a nose circuit of \mathcal{S} . Conversely, if \mathcal{W} is a nose circuit, and $B'_1 \rightarrow B_1, \dots, B'_a \rightarrow B_a$ are its noses, then $\mathcal{A}(\mathcal{W}) = B_1, \dots, B_a$ is an standard (a, b) -independent for some b . It is not hard to see that $\mathcal{A}(\mathcal{W}(\mathcal{A})) = \mathcal{A}$ and $\mathcal{W}(\mathcal{A}(\mathcal{W})) = \mathcal{W}$, thus the correspondence is one-to-one.

Observe that the number of turns b in the *ss traversal* of \mathcal{A} is precisely the number of external noses and steps of $\mathcal{W} = \mathcal{W}(\mathcal{A})$. In other words,

$$b = \nu_{-h} + \nu_{1-h} + \sigma_{-h}.$$

Similarly, the number a of arcs of \mathcal{A} equals the number of noses of \mathcal{W} ; by (1),

$$a = h(\nu_{-h} + \nu_{1-h} + \sigma_{-h}) + \nu_{-h} - \sigma_1 = hb + \nu_{-h} - \sigma_1.$$

Hence, $a/b = h + r(\mathcal{W})$.

A similar analysis holds for (x, y) -circuits. Say that an (x, y) -circuit $\mathcal{A} = B_1, \dots, B_x$ is *standard* when $s(B_i)$ immediately precedes an ending point $t(B'_i)$ in \mathcal{M} . Note that \mathcal{M} contains an (x, y) -circuit if and only if it contains an standard (x, y) -circuit; in such circuit, $B_i \rightarrow B'_i$ is a hollow of \mathcal{S} . Then, \mathcal{A} is in a one-to-one correspondence with the hollow circuit $\mathcal{W} = \mathcal{W}(\mathcal{A})$ that goes through $\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_x$ where each \mathcal{W}_i is the step path going from B'_i to B_{i+1} . As before, the number of turns in the *ss traversal* of \mathcal{A} is the number of external hollows minus the number of external steps, i.e.,

$$y = \eta_h + \eta_{h-1} - \sigma_{-h},$$

while the number x of arcs in \mathcal{A} is its number of hollows; by (1)

$$x = h(\eta_h + \eta_{h-1} - \sigma_{-h}) + \eta_h + \eta_0 + \sigma_1.$$

Hence, $x/y = h + R(\mathcal{W})$.

Clearly, we can obtain $\mathcal{W}(\mathcal{A})$ in $O(n)$ time for any standard (a, b) -independent (resp. (x, y) -circuit) \mathcal{A} , and vice versa. Moreover, note that, as stated, $a/b \geq x/y$ if and only if $r \geq R$. We summarize this section in the next theorem.

Theorem 14. *A PCA model \mathcal{M} contains an (a, b) -independent (resp. (x, y) -circuit) \mathcal{A} if and only if \mathcal{S} contains a nose (resp. hollow) circuit \mathcal{W} with ratio $a/b - h$ (resp. $x/y - h$). Moreover, such a circuit \mathcal{W} of \mathcal{S} can be obtained in $O(n)$ time when \mathcal{A} is given as input. Conversely, \mathcal{A} can be obtained in $O(n)$ time when \mathcal{W} is given as input.*

8 Minimal UCA and UIG models

Theorem 1 gives us a procedure to check if \mathcal{M} is equivalent to a u -CA model, when u is given as input. However, not much is known about the sets of feasible values c and ℓ . In this aspect, unit circular-arc models are much less studied than unit interval models. In this section we prove that every UCA model admits an equivalent *minimal* UCA model. Minimal UCA models are a generalization of minimal UIG models, as defined by Pirlot [28]. An (ℓ, d, d_s) -IG model with arcs $A_1 < \dots < A_n$ is (d, d_s) -minimal when

(min-uig₁) $\ell \leq \ell'$, and

(min-uig₂) $s(A_i) < s(A'_i)$ for every $1 \leq i \leq n$,

for every equivalent (ℓ', d, d_s) -IG model.

Condition (min-uig₂) as expressed above does not make much sense for UCA models, as there is not a natural left-to-right order of the arcs; the 0 point of the circle is just a denotational tool. However we can translate condition $s(A_n) < s(A'_n)$ by asking the circumference of the circle to be minimized. With this in mind, we say that a (c, ℓ, d, d_s) -CA model \mathcal{M} is (d, d_s) -minimal when

(min-uca₁) $\ell \leq \ell'$, and

(min-uca₂) $c \leq c'$,

for every equivalent (c', ℓ', d, d_s) -CA model. The fact that every UCA model is equivalent to a minimal UCA model follows from the next lemma.

Lemma 15. *If \mathcal{M} is equivalent to a $(c, \ell + y, d, d_s)$ -CA model and a $(c + x, \ell, d, d_s)$ -CA model for $x, y \geq 0$, then \mathcal{M} is also equivalent to a $(c + a, \ell + b, d, d_s)$ -CA model, for every $0 \leq a \leq x$ and $0 \leq b \leq y$.*

Proof. For the sake of notation, write $\langle a', b' \rangle$ to denote the UCA descriptor $(c + a, \ell + b, d, d_s)$, for every $0 \leq a' \leq x$ and $0 \leq b' \leq y$. Suppose, to obtain a contradiction, that \mathcal{M} is equivalent to no $\langle a, b \rangle$ -CA model for some $1 \leq a \leq x$ and $1 \leq b \leq y$. Then, by Theorem 1, $\text{sep}_{\langle a, b \rangle}(\mathcal{W}) > 0$, $\text{sep}_{\langle x, 0 \rangle}(\mathcal{W}) \leq 0$, and $\text{sep}_{\langle 0, y \rangle}(\mathcal{W}) \leq 0$ for some cycle \mathcal{W} of \mathcal{S} . By (4), there exists $e \geq 0$ such that

$$c + a' = (\ell + y + d)(h + r) + e + a' = (\ell + b' + d)(h + r) + (y - b')(h + r) + e + a' \quad (\text{i})$$

for every $0 \leq a' \leq x$ and $0 \leq b' \leq y$. Thus, by (5),

$$\text{sep}_{\langle a, b \rangle} = (\ell + b + d)\text{len} + ((y - b)(h + r) + e + a)\text{ext} + \text{const} > 0 \quad (\text{ii})$$

$$\text{sep}_{\langle 0, y \rangle} = (\ell + y + d)\text{len} + e\text{ext} + \text{const} \leq 0 \quad (\text{iii})$$

$$\text{sep}_{\langle x, 0 \rangle} = (\ell + d)\text{len} + (y(h + r) + e + x)\text{ext} + \text{const} \leq 0. \quad (\text{iv})$$

Recall that $\text{len} < 0$ by Theorem 6. Then, as $0 < (\text{ii}) - (\text{iv})$, we obtain that $(\text{v}) \text{ ext} \leq -1$ (recall $\text{ext} \in \mathbb{Z}$), while $(\text{vi}) (h+r)\text{ext} > \text{len}$ follows by $0 < (\text{ii}) - (\text{iii})$ and (v) . Then,

$$\begin{aligned}
0 &< (\ell + b + d)\text{len} + ((y - b)(h + r) + e + a)\text{ext} + \text{const} && (\text{by (ii)}) \\
&< (\ell + b + d)(h + r)\text{ext} + ((y - b)(h + r) + e + a)\text{ext} + \text{const} && (\text{by (vi)}) \\
&= (c + a)\text{ext} + \text{const} && (\text{by (i)}) \\
&\leq -c - a + \text{const} && (\text{by (v)})
\end{aligned}$$

This is impossible, because $c \geq \max\{2d, d + d_s\}n$ as all the extremes of the $\langle 0, y \rangle$ -CA model equivalent to \mathcal{M} are separated by d and each of its n beginning points is separated from the next by $d + d_s$, while $a \geq 0$ and $\text{const} \leq \max\{2d, d + d_s\}n$ by definition (8). \square

Theorem 16. *Every UCA graph admits a (d, d_s) -minimal UCA model for every $d, d_s \in \mathbb{Q}$.*

For the rest of this section, we restrict ourselves to the case in which d and d_s are integers. By (2), $r = 0$ for every UIG model \mathcal{M} , thus, by (5), $\text{sep}(\mathcal{W}) \in \mathbb{N}$ if and only if ℓ and c are integers. We obtain, therefore, the following corollary that was first proved by Pirlot [28].

Corollary 17. *Every (d, d_s) -minimal UIG model is integer for every $d, d_s \in \mathbb{N}$*

The above corollary holds for UCA models with an integer nose ratio as well. However, we were not able to prove or disprove the above corollary for the general case. For this reason, we say that an integer (c, ℓ, d, d_s) -CA model \mathcal{M} is (\mathbb{N}, d, d_s) -minimal when it satisfies (min-uca₁) and (min-uca₂) for every integer (c', ℓ', d, d_s) -CA model.

A natural algorithmic problem is MINUCA in which we ought to find a (\mathbb{N}, d, d_s) -minimal UCA model equivalent to an input (c, ℓ, d, d_s) -CA model \mathcal{M} . A simple solution is to apply Theorem 1 for every $1 \leq \ell^* \leq \ell$ and every $1 \leq c^* \leq c$ with a total cost of $O(\ell^* c^* n^2)$ time. We can easily improve this algorithm by replacing the linear search of c^* with a binary search.

Corollary 18. *Let \mathcal{M} be a PCA model. If $\text{sep}_{(c, \ell, d, d_s)}(\mathcal{W}) > 0$ for some cycle \mathcal{W} of \mathcal{S} , then \mathcal{M} is not equivalent to a $(c + \text{sg}.x, \ell - y, d, d_s)$ -CA model, for every $x, y \geq 0$, where sg is the sign of $\text{ext}(\mathcal{W})$.*

Proof. Let e be such that $c = (\ell + d)(h + r) + e$; note that e needs not be positive. By (5),

$$\text{sep}_{(c + \text{sg}.x, \ell, d, d_s)} = (\ell + d)\text{len} + (e + \text{sg}.x)\text{ext} + \text{const} = \text{sg}.x.\text{ext} + \text{sep}_{(c, \ell, d, d_s)} > 0.$$

Then, by Theorem 1, \mathcal{M} cannot be equivalent to a $(c + \text{sg}.x, \ell, d, d_s)$ -CA model. \square

Corollary 18 provides us with a somehow efficient algorithm to binary search the minimum $c^* \in \mathbb{N}$ such that \mathcal{M} is equivalent to a (c^*, ℓ, d, d_s) -CA model, when $\ell, d, d_s \in \mathbb{N}$ are given as input. By definition $d < \ell$, while we assume $d_s + d < \ell$ as otherwise $G(\mathcal{M})$ has no edges and the problem is trivial. The idea of the algorithm is simply to assume that such a value c^* exists and belongs to some range $[a, c]$; initially $a = 0$ and $c = n(\ell + 1)$. Then, we query if \mathcal{M} is equivalent to some (b, ℓ, d, d_s) -CA model, where $b \in \mathbb{N}$ is the middle of $[a, c]$. If affirmative, then $c^* \in [a, b]$ by definition. Otherwise, we search some cycle \mathcal{W} with $\text{sep}_{(b, \ell, d, d_s)} > 0$. By Corollary 18, $c^* \in [0, b]$ if $\text{ext}(\mathcal{W}) \geq 0$, while $c^* \in (b, c]$ otherwise. Regardless of whether c^* exists or not, this algorithm requires $O(\log(n\ell))$ queries.

Every time we need to query if \mathcal{M} is equivalent to a (b, ℓ, d, d_s) -CA model, we solve u -REP as in Section 3.1. Since $\ell^* = O((d + d_s)n^2)$ [24], we conclude that the total time required to obtain (c^*, ℓ^*) , thus solving MINUCA, is $O((d + d_s)n^4 \log(n + d + d_s))$.

8.1 Minimal UIG models

Piriot proved in [28] that every UIG model \mathcal{M} is equivalent to a $(1,0)$ -minimal UIG model. However, it was Mitas who showed that such a model can be found in linear time by transforming \mathcal{M} into an equivalent UIG model \mathcal{M}^* [27]. Unfortunately, her proof has a flaw that invalidates the minimality arguments. Though \mathcal{M}^* is equivalent to \mathcal{M} , it needs not be $(1,0)$ -minimal. On the other hand, the algorithm in the previous section can be implemented so as to run in $O(n^2 \log n)$ time when applied to \mathcal{M} . (Below we discuss condition why [\(min-uig₂\)](#) is satisfied by the unit interval model so obtained.) In this section we briefly describe Mitas' algorithm and its counterexample, and propose a patch. The obtained algorithm runs in $O(n^2)$ time, and its a compromise between Mitas' algorithm and the algorithm in the previous section.

Let \mathcal{M} be a PIG model with arcs $A_1 < \dots < A_n$. By definition, no arc of \mathcal{M} crosses 0, thus \mathcal{S} has no external hollows. Similarly, external noses and steps are redundant in \mathcal{S} for testing if \mathcal{M} is equivalent to a ℓ -IG model, as $c = \infty$. Therefore, we assume that \mathcal{S} has only five types of edges, namely 1-noses, 0- and 1-steps, and 0- and (-1) -hollows. Mitas identifies two special vertices of \mathcal{S} for each height value. A *leftmost* vertex is a vertex A_i such that either $i = 1$ or $h(A_i) = h(A_{i-1}) + 1$, while a *rightmost* vertex is a vertex A_j such that either $j = n$ or $h(A_{j+1}) = h(A_j) + 1$.

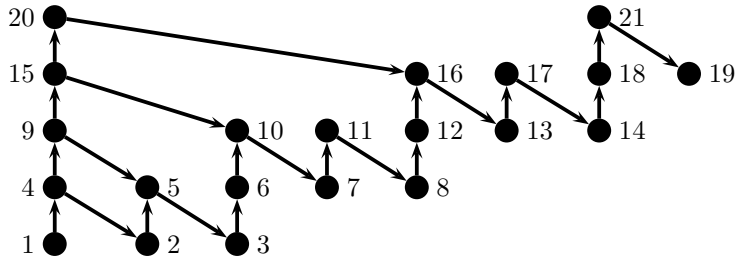
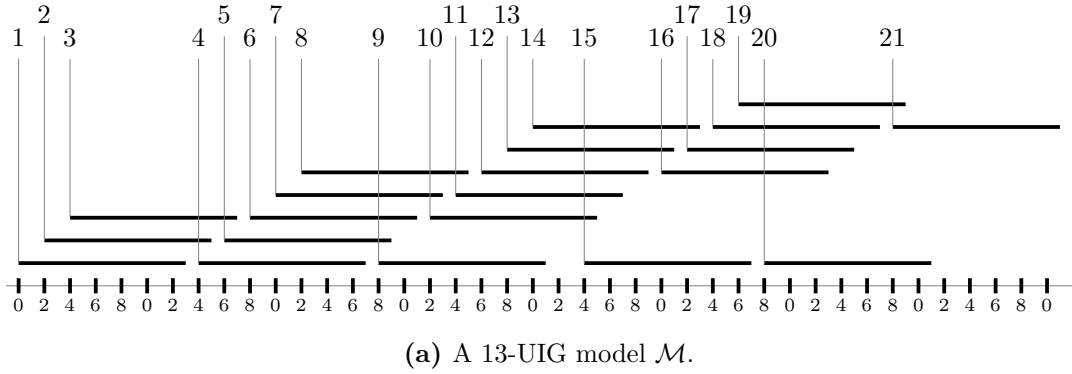


FIGURE 2. Counterexample to (11): $\text{dsep}(\mathcal{T}, A_1, A_{19}) - \text{dsep}(\mathcal{T}, A_1, A_{15}) = 14$ while in \mathcal{S} the maximum cycle has length 13.

Suppose \mathcal{M} is equivalent to some ℓ -IG model and let $\mathcal{U} = \mathcal{U}(\infty, \ell)$ be as defined in Theorem 1, but replacing \mathcal{B} and A_0 with \mathcal{S} and A_1 , respectively. That is, the arc U_i corresponding to A_i begins at $s(U_i) = \text{dsep}(A_1, A_i)$ for every $1 \leq i \leq n$. For $x \in \{1, \dots, h\}$, let A_i and A_j be leftmost and rightmost vertices with $h(A_i) = h(A_j) = x$, and \mathcal{W}_x be a path from A_i to A_j . By definition of \mathcal{U} , it follows that $s(U_j) - s(U_i) \geq \text{sep}(\mathcal{W}_x)$. That is,

$$\text{dsep}(A_1, A_j) - \text{dsep}(A_1, A_i) \geq \text{sep}(\mathcal{W}_x). \quad (10)$$

Mitas' key idea is to take ℓ so that (10) holds by equality when $\text{sep}(\mathcal{W}_x)$ is maximum (recall sep depends on ℓ). The flaw, however, is that she discards 0-hollows and 1-steps before solving (10).

To make the above statement more precise, let $\mathcal{T}(\mathcal{M})$ be the digraph obtained from $\mathcal{S}(\mathcal{M})$ by removing all the 0-hollows and 1-steps, and $\mathcal{W}_x^\mathcal{T}$ be a path from A_i to A_j in \mathcal{T} (as usual we drop the parameter \mathcal{M} from \mathcal{T}). Mitas claims (Theorem 5 in [27], although using a different terminology) that

$$\mathbf{dsep}(\mathcal{T}, A_1, A_j) - \mathbf{dsep}(\mathcal{T}, A_1, A_i) = \text{sep}(\mathcal{W}_x^\mathcal{T}). \quad (11)$$

when $\text{sep}(\mathcal{W}_x^\mathcal{T})$ is maximum and ℓ is minimum. Figure 2 shows a counterexample to this fact. The inconvenient is that $\mathbf{dsep}(\mathcal{S}, A_1, A_i)$ is greater than $\mathbf{dsep}(\mathcal{T}, A_1, A_i)$ when every maximum path from A_1 to A_i contains the 0-hollow or 1-step ending at A_i .

Equation (11) is fundamental for keeping the time and space complexity low. The main observation is that \mathcal{T} is acyclic whereas \mathcal{S} is not.

Lemma 19 ([27]). *Digraph $\mathcal{T}(\mathcal{M})$ is acyclic, for any UIG model \mathcal{M} .*

Given that \mathcal{T} is acyclic, we can compute the *column* that every arc of \mathcal{M} occupies in a pictorial description of \mathcal{T} . The *column* of A_1 is $c(A_1) = 0$, while, for every $1 < i \leq n$, the *column* of A_i is:

$$c(A_i) = \max \left\{ \begin{array}{l} c(N) + \varepsilon \\ c(H) + 1 \\ c(S) + 1 \end{array} \left| \begin{array}{l} N \rightarrow A_i \text{ is a 1-nose} \\ H \rightarrow A_i \text{ is a 0-step} \\ S \rightarrow A_i \text{ is a } -1\text{-hollow} \end{array} \right. \right\} \quad (12)$$

for a small enough ε (say $\varepsilon \ll 1/n$); obviously, if A_i is not the end of a nose (resp. hollow, step), then the corresponding value in the above equation is 0. It is easy to see, by the existence of 0-steps, that $c(I_i) \leq c(I_k) \leq c(I_j)$ when A_i and A_j are the leftmost and rightmost with height x , for every $A_i < A_k < A_j$. In Figure 2, each vertex A of \mathcal{S} occupies the coordinate $(c(A), h(A))$ on the plane, for some imperceptible ε , while each directed edge is a straight arrow. This pictorial description, which we call the *canonical drawing* of \mathcal{T} , was proposed by Mitas and it is quite useful for simplifying some geometrical arguments. The reason is that this drawing is a plane digraph; we include a proof of this fact as it is not completely explicit in [27].

Theorem 20 (see [27, 30]). *The canonical drawing of \mathcal{T} is a plane digraph.*

Proof. Suppose, to obtain a contradiction, that the canonical drawing of \mathcal{T} it is not a plane graph. Then, there are two crossing straight lines that correspond to the edges $A_i \rightarrow A_j$ and $A_x \rightarrow A_y$ with $h(A_i) \leq h(A_x)$. By definition, \mathcal{T} has only 1-noses, (-1) -hollows and 0-steps, while every vertex A is positioned in $(c(A), h(A))$. Hence, it follows that $A_i \rightarrow A_j$ is a 1-nose, $A_x \rightarrow A_y$ is a (-1) -hollow, and $A_i < A_y < A_x < A_j$. But this configuration is impossible because it implies that $t(A_i)s(A_j)$ and $s(A_x)t(A_y)$ are consecutive, while $t(A_i) < t(A_y)$ and $s(A_x) < s(A_j)$. \square

Corollary 21 (Theorem 11). *If \mathcal{M} is a PCA graph, then \mathcal{S} is a toroidal digraph.*

Proof. A torus can be obtained from a rectangle by first pasting its north and south borders together, and then pasting the east end of the obtained cylinder with its west end. Thus, it suffices to show how to draw \mathcal{S} into a rectangle allowing some edges to escape from the north (resp. east) into the south (resp. west). Let \mathcal{T} be obtained from \mathcal{S} by removing all the external edges, plus 1-steps and 0-hollows. To draw \mathcal{S} , first copy the canonical drawing of \mathcal{T} into the rectangle. Then, draw all the 0-hollows and 1-steps so that they escape through the east, all the h -hollows and $(-h)$ -noses so as to run through the north, and the $(-h)$ -step and all the $(h-1)$ -hollows and $(1-h)$ -noses by going through the north first and then through the east. It is not hard to see that such a drawing is always possible. \square

In the next lemma we take advantage of the canonical drawing to prove that every cycle of \mathcal{S} contains exactly one 0-hollow or 1-step. Pirlot also studies the shape of the cycles of \mathcal{S} [28], but without taking advantage of Mitas' canonical drawing. For the next lemma, recall that $\text{len}(\mathcal{W}) = -\eta_0(\mathcal{W}) - \sigma_1(\mathcal{W})$ for any cycle \mathcal{W} .

Lemma 22 ([28, Proposition 2.11]). *If \mathcal{M} is a PIG model, then $\text{len}(\mathcal{W}) = -1$ for any cycle \mathcal{W} of \mathcal{S} .*

Proof. Note that $\text{len} = -\eta_0 - \sigma_1$ (6), hence, by Lemma 19, $\text{len} < 0$. Suppose, to obtain a contradiction, that $\text{len} < -1$. Then, \mathcal{W} has a subpath B_1, \dots, B_j with no 0-hollows nor 1-steps such that $B_0 \rightarrow B_1$ and $B_j \rightarrow B_{j+1}$ each is either a 0-hollow or a 1-step of \mathcal{W} . Among all such possible paths, take \mathcal{W} so that $h(B_1)$ is maximum. Note that $B_0 \neq B_j$, thus \mathcal{W} has another path B_{-k}, \dots, B_0 such that B_{-k} is its unique leftmost vertex. By the maximality of $h(B_1)$, it follows that $h(B_0) \geq h(B_1) - 1 \geq h(B_{-k})$ while, since $h(B_j) \leq h(B_{j+1}) \leq h(B_1) - 1 \leq h(B_0)$ and $B_0 \neq B_j$, it follows that $h(B_0) > h(B_j)$.

Call Gr^+ to the curve that results by traversing B_1, \dots, B_j in the canonical drawing of \mathcal{T} . Note that Gr^+ is indeed the graph of a continuous function on $\mathbb{R} \rightarrow \mathbb{R}$ because $c(B_{i+1}) > c(B_i)$ for every $1 \leq i < j$ by (12). Similarly, the curve Gr^- that results by traversing B_{-k}, \dots, B_0 in the canonical drawing of \mathcal{T} is also the graph of a continuous function. Since $h(B_{i+1}) = h(B_i) \pm 1$ for every $i \in \{-k, \dots, j\} \setminus 0$, it follows that Gr^+ contains a vertex with height x for every $h(B_j) \leq x \leq h(B_1)$ and Gr^- contains a vertex with height x for every $h(B_{-k}) \leq x \leq h(B_0)$. Then, taking into account that B_1 and B_{-k} are leftmost vertices with $h(B_1) > h(B_{-k})$ and B_j and B_0 are rightmost vertices with $h(B_0) > h(B_j)$, we obtain that Gr^+ and Gr^- intersect. Hence, by Theorem 20, B_1, \dots, B_j and B_{-k}, \dots, B_0 have a nonempty intersection, which implies that \mathcal{W} is not a cycle. \square

By Lemma 22 and (5), $\text{sep}_u(\mathcal{W}) = \text{const}(\mathcal{W}, d, d_s) - \ell - 1$ for every cycle \mathcal{W} . Then, by Theorem 1 and Lemma 22, the minimum ℓ^* such that \mathcal{M} is equivalent to an (ℓ^*, d, d_s) -IG model is

$$\begin{aligned} \ell^* + 1 &= \max\{\text{const}(\mathcal{W}, d, d_s) \mid \mathcal{W} \text{ is a cycle of } \mathcal{S}\} \\ &= \max\{\text{const}(\mathcal{W}, d, d_s) \mid \mathcal{W} \text{ is a path } B_1, \dots, B_j \text{ of } \mathcal{T} \text{ for a 1-step or 0-hollow } B_j \rightarrow B_1\} \\ &= \max\{\mathbf{dconst}_{d, d_s}(\mathcal{T}, A_i, A_j) \mid A_j \rightarrow A_i \text{ is either a 1-step or 0-hollow}\}. \end{aligned}$$

Since \mathcal{T} is acyclic, we can compute $\mathbf{dconst}_{d, d_s}(\mathcal{T}, A_i, A_j)$ in $O(n)$ time and space for any given 1-step or 0-hollow $A_j \rightarrow A_i$ of \mathcal{S} . Then, ℓ^* is obtained in $O(hn)$ time.

Once ℓ^* has been obtained, $\mathcal{U} = \mathcal{U}(\infty, \ell^*, d, d_s)$ can be constructed in $O(n^2)$ time and linear space as in Section 3.1. We claim that \mathcal{U} is a (d, d_s) -minimal UIG model. Indeed, \mathcal{U} satisfies (min-uig₁) by the minimality of ℓ^* . To see that \mathcal{U} satisfies (min-uig₂), consider any path \mathcal{W} of \mathcal{S} from A_1 to A_j . Note that $h(A_j) \geq \sigma_1 + \eta_0 = -\text{len}$ because no leftmost vertex is traversed twice by \mathcal{W} . Therefore, by (5),

$$\text{sep}_{(\infty, \ell, d, d_s)} = (\ell + 1)(h(A_j) + \text{len}) + \text{const} \geq (\ell^* + 1)(h(A_j) + \text{len}) + \text{const} = \text{sep}_{(\infty, \ell^*, d, d_s)}$$

for any $\ell \geq \ell^*$. Consequently, since $s(A_j) \geq \mathbf{dsep}_{(\infty, \ell, d, d_s)}(A_1, A_j)$ in any (ℓ, d, d_s) -UIG model equivalent to \mathcal{U} , it follows that \mathcal{U} satisfies (min-uig₂) as well. We conclude that $O(n^2)$ time and linear space suffices to solve the *minimal UIG representation* (MINUIG) problem in which \mathcal{M} and $d, d_s \in \mathbb{Q}_{\geq 0}$ are given and a (d, d_s) -minimal UIG model equivalent to \mathcal{M} must be generated.

Theorem 23. *MINUIG can be solved in $O(n^2)$ time and linear space, for any $d, d_s \in \mathbb{Q}$.*

9 Powers of paths and cycles

Powers of paths and cycles are intimately related to UIG and UCA graphs, respectively. For any graph G , its k -th power G^k is the graph obtained from G by adding an edge between v and w whenever there is a path in G of length at most k joining them. In this section we write P_q and C_q to denote the path and cycle graphs with q vertices, respectively. Lin et al. [26] noted that G is a UCA (resp. UIG) graph if and only if G is an induced subgraph of C_q^k (resp. P_q^k) for some q, k (see also [9] for UIG graphs and [11] for UCA graphs).

In [7], Costa et al. propose a specialized $O(n^2)$ time and space algorithm whose purpose is to find the minimum values k and $q(k)$ such that a UIG graph G is an induced subgraph of $P_{q(k)}^k$. The reason for writing $q(k)$ dependent on k is to be as truthful to [7] as we can; they always write the number of vertices as a function on the power. This is not important, though, as we know that q is independent of k by Pirlot's minimality Theorem [28]. That is, q is the minimum such that G is an induced subgraph of P_q^k for every possible k . Mitás' algorithm could have been applied to obtain k and q in $O(n)$ time and space, under the assumption that it is correct. Interestingly, Pirlot's Theorem and Mitás' algorithm predate [7] for at least fifteen years. Moreover, [32, Section 9], which is referenced within [7], mentions that Mitás' algorithm could be adapted to work when the input is a PIG model. The purpose of this section is to apply the minimization algorithms so as to find powers of paths and cycles supergraphs.

Let C_q^k (resp. P_q^k) be the $(2q, 2k+1)$ -CA (resp. $(2k+1)$ -IG) model that has an arc with beginning point $2i$ for every $0 \leq i < q$. It is not hard to see that C_q^k (resp. P_q^k) is a $(1, 0)$ - and $(1, 1)$ -minimal model representing C_q^k (resp. P_q^k). We say that a (c, ℓ) -CA (resp. ℓ -IG) model \mathcal{M}^* is *completable* when \mathcal{M}^* can be obtained by removing arcs from C_q^k (resp. P_q^k) for some $k, q \geq 0$. In such case, C_q^k (resp. P_q^k) is referred to as the *completion* of \mathcal{M}^* , while \mathcal{M}^* is said to be a (k, q) -*extension* of \mathcal{M} for every UCA (resp. UIG) model \mathcal{M} equivalent to \mathcal{M}^* . Note that \mathcal{M}^* is completable if and only if:

(ext₁) ℓ is odd,

(ext₂) c is even, and

(ext₃) all its beginning points are even (thus \mathcal{M}^* is a $(c, \ell, 1, 1)$ -CA model).

Under this new terminology, the result by Lin et al. [26] states that every UCA (resp. UIG) model \mathcal{M} admits a (k, q) -extension \mathcal{M}^* for some $k, q \geq 0$. In analogy to minimal models, we say that \mathcal{M}^* is a *minimal extension* of \mathcal{M} when $q \leq q'$ and $k \leq k'$ for every (k', q') -extension of \mathcal{M} . The *minimal power of a cycle (resp. path)* MINC_q^k (resp. MINP_q^k) problem consists of finding \mathcal{M}^* when the UCA (resp. UIG) model \mathcal{M} is given as input. A priori, \mathcal{M} could have no minimal extensions. But, if \mathcal{M}^* is the minimal extension of \mathcal{M} , then, clearly, k and q are the minimum values such that $G(\mathcal{M})$ is an induced subgraph of C_q^k (resp. P_q^k). We now discuss how to solve MINC_q^k and MINP_q^k .

The fact that \mathcal{M} admits a minimal extension follows by Lemma 15 and Theorem 1. Indeed, if ℓ^* is the minimum odd number such that \mathcal{M} is equivalent to a $(c, \ell^*, 2)$ -CA model, and c^* is the minimum even number such that \mathcal{M} is equivalent to a $(c^*, \ell, 1, 1)$ -CA model, then, by Lemma 15, \mathcal{M} is equivalent to a $(c^*, \ell^*, 1, 1)$ -CA model. Furthermore, $\text{sep}_{c^*, \ell^*, 1, 1}(A_i \rightarrow A_j)$ is even for every edge $A_i \rightarrow A_j$ of \mathcal{S} , by (sep₁)–(sep₄). Thus, all the beginning points of $\mathcal{M}^* = \mathcal{U}(\mathcal{M}, c^*, \ell^*, 1, 1)$ are even. Then, \mathcal{M}^* is completable by (ext₁)–(ext₃), while it is equivalent to \mathcal{M} by Theorem 1. That is, \mathcal{M}^* is the minimal extension of \mathcal{M} and, thus, the solution to MINC_q^k . The values ℓ^* and c^* can be found $O(n^4 \log n)$ time with an algorithm similar to the one in Section 8.

For the special case in which \mathcal{M} is a UIG model, we observe that any $(1, 1)$ -minimal model \mathcal{M}^* equivalent to \mathcal{M} is a minimal extension of \mathcal{M} . Just recall that the length ℓ^* of the arcs in

\mathcal{M}^* is equal to $\text{const}(\mathcal{W}, 1, 1) - 1$ for some path \mathcal{W} of $\mathcal{S}(\mathcal{M}^*)$. Since $\text{const}(\mathcal{W}, 1, 1)$ is even, it follows that ℓ^* is odd and, thus, $\text{sep}_{\infty, \ell^*, 1, 1}(A_i \rightarrow A_j)$ is even for every edge $A_i \rightarrow A_j$ of $\mathcal{S}(\mathcal{M}^*)$. By (ext₁)–(ext₃), this implies that \mathcal{M}^* is an extension of \mathcal{M} which, of course, is minimal by (min-[uig₁](#)) and (min-[uig₂](#)). By Theorem 23, MINP_q^k is solvable in $O(n^2)$ time and linear space.

10 Further remarks

Synthetic graphs proved to be an important tool for studying how do the UIG representations of PIG graphs look like. The generalization to PCA models is direct; the fact that some arcs wrap around the circle is not important for defining the synthetic graph. To represent the separation constraints that an equivalent UCA model must satisfy, all we had to include to Pirlot’s original formulation was the variable c representing the circumference of the circle. Generalizations of simple ideas from PIG to PCA graphs are not always as easy to obtain. Unfortunately, Pirlot’s ideas were introduced in the context of semiorders and were not exploited in the context of PCA graphs; the recognition problem of UCA graphs in polynomial time could have been solved more than a decade earlier. In this closing section we provide some remarks and discuss some open problems.

Our definition of UCA descriptors states that every pair of beginning points should be separated by $d + d_s$ distance. An obvious generalization to u -REP and (INT)BOUNDREP is to replace d_s with a function $d_s: \mathcal{A} \rightarrow \mathbb{Q}_{\geq 0}$ that indicates, for each arc A_i , the separation between $s(A_i)$ and the next beginning point $s(A_{i+1})$. The reader can check that Theorem 1 holds for this generalization as well. All we need to do is to replace the value d_s with $d_s(A_i)$ for each step $A_i \rightarrow A_{i+1}$. Moreover, we can use similar functions to further separate $t(A_i)$ from $s(A_j)$ for every nose $A_i \rightarrow A_j$, and $s(A_i)$ from $t(A_j)$ for any hollow $A_i \rightarrow A_j$. We did not consider these generalization for the sake of simplicity and notation.

In Section 8 we gave a simple polynomial algorithm to transform a UCA model \mathcal{M} into a minimal (c^*, ℓ^*, d, d_s) -CA model. The algorithm works by performing a linear search on ℓ^* and a binary search on c^* . An obvious idea to improve its running time is to replace the linear search on ℓ^* with a binary search. Unfortunately, this idea is not feasible at first sight because we cannot claim

$$L = \{\ell \in \mathbb{N} \mid \mathcal{M} \text{ is equivalent to a } (c, \ell, d, d_s)\text{-CA model for some } c \in \mathbb{N}\}$$

to be a range. For instance, C_{11}^4 admits a $(22, 9)$ -CA model, but it admits no $(c, 10)$ -CA model, whatever value of c is. This is just one more example of a property that is lost when the linear structure of PIG models is replaced by the circular structure of PCA graphs as $L = [\ell^*, \infty)$ when \mathcal{M} is PIG.

As calculated in Section 8, the running time of the minimization algorithm is $O((d + d_s)n^4 \log(n + d + d_s))$. This bound is not tight, as the actual running time is $O(\ell^* n^2 \log(n \ell^*))$, and ℓ^* could be much lower than $(d + d_s)n^2$. As a matter of fact, we developed a simple program for testing if a UCA model is equivalent to some $(c, 2n)$ -CA model. We tested it on many input UCA models and, in all cases, the program was successful.

In Section 8.1 we fixed Mitas’ algorithm so as to solve the minimization problem for UIG models. Unfortunately, the running time of the patched algorithm is $O(n^2)$. There are two bottlenecks in this algorithm. First, we have to compute the minimum length value ℓ^* . Then, we have to apply the Bellman-Ford algorithm to compute the actual model. With respect to the space complexity, it is not hard to observe that ℓ^* can be computed in unambiguous logspace. Indeed, all we have to do is to find the distance between the leftmost and rightmost arcs for every height. As the canonical drawing is a plane graph with $O(1)$ vertices with 0 in-degree,

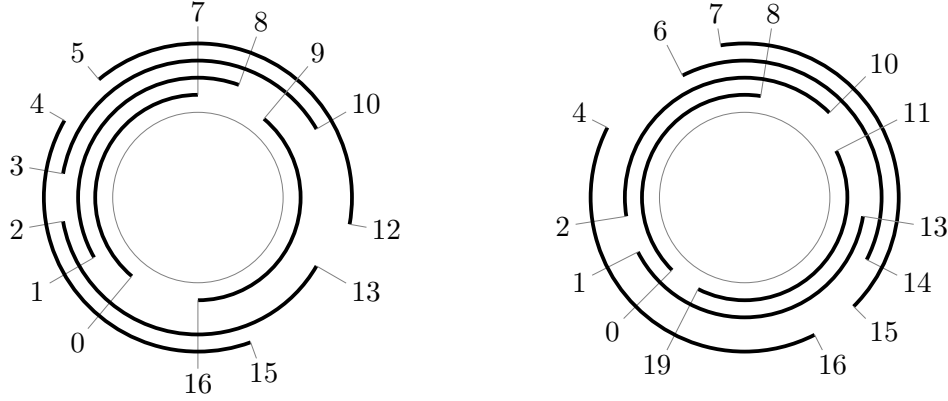


FIGURE 3. Two minimal UCA models representing the same graph.

this problem requires unambiguous logspace [23]. Finding logspace algorithms to compute ℓ^* and the minimal model remain as open problems.

Finally, it should be noted that a UCA graph may admit many *non-equivalent* minimal UCA models. Indeed, a UCA graph may admit an exponential number of non-equivalent models, each of which is equivalent to a minimal UCA model. It makes sense, then, to say that a model \mathcal{M} is *minimum* when it satisfies (min-uca₁) and (min-uca₂) for every model \mathcal{M}' such that $G(\mathcal{M})$ is isomorphic to $G(\mathcal{M}')$. As it was noted by Huang [14], every connected and co-connected PCA graph admits a unique PCA model, up to equivalence and full reversal. Thus, any minimal model of a connected and co-connected PCA graph is minimum. Similarly, every disconnected PCA graph is PIG, and all its models can be obtained from a model \mathcal{M} by exchanging the order in which their components appear from 0, and reversing some of the components. Thus, again, any minimal model of $G(\mathcal{M})$ is minimum. Co-disconnected PCA graphs share a similar property: all their PCA models can be obtained from a PCA model \mathcal{M} by exchanging the order in which its co-components appear, plus reversing some co-components [14]. Thus, one is tempted to think that all the minimal PCA models are minimum, yet this is not the case. Figure 3 shows two 1-minimal (18, 7)-CA and (20, 8)-CA models that represent the graph whose co-components are $P_2 \cup P_1$ and P_4 . We leave as open the problem of computing the minimum UCA model.

References

- [1] M. Balko, P. Klavík, and Y. Otachi. Bounded representations of interval and proper interval graphs. In L. Cai, S.-W. Cheng, and T.-W. Lam, editors, *Algorithms and Computation*, vol. 8283 of *Lecture Notes in Computer Science*, pp. 535–546. Springer, 2013. doi:[10.1007/978-3-642-45030-3_50](https://doi.org/10.1007/978-3-642-45030-3_50).
- [2] T. Bläsius and I. Rutter. Simultaneous pq-ordering with applications to constrained embedding problems. In S. Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pp. 1030–1043. SIAM, 2013. doi:[10.1137/1.9781611973105.74](https://doi.org/10.1137/1.9781611973105.74).
- [3] S. Chaplick, P. Dorbec, J. Kratochvíl, M. Montassier, and J. Stacho. Contact representations of planar graph: Rebuilding is hard. In *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Comput. Sci. Springer, Berlin.
- [4] S. Chaplick, R. Fulek, and P. Klavík. Extending partial representations of circle graphs. In S. Wismath and A. Wolff, editors, *Graph Drawing*, vol. 8242 of *Lecture Notes in Computer Science*, pp. 131–142. Springer, 2013. doi:[10.1007/978-3-319-03841-4_12](https://doi.org/10.1007/978-3-319-03841-4_12).

- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.
- [6] D. G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A. P. Sprague. Simple linear time recognition of unit interval graphs. *Inform. Process. Lett.*, 55(2):99–104, 1995. doi:[10.1016/0020-0190\(95\)00046-F](https://doi.org/10.1016/0020-0190(95)00046-F).
- [7] V. Costa, S. Dantas, D. Sankoff, and X. Xu. Gene clusters as intersections of powers of paths. *J. Braz. Comput. Soc.*, 18(2):129–136, 2012. doi:[10.1007/s13173-012-0064-8](https://doi.org/10.1007/s13173-012-0064-8).
- [8] G. Durán, A. Gravano, R. M. McConnell, J. Spinrad, and A. Tucker. Polynomial time recognition of unit circular-arc graphs. *J. Algorithms*, 58(1):67–78, 2006. doi:[10.1016/j.jalgor.2004.08.003](https://doi.org/10.1016/j.jalgor.2004.08.003).
- [9] N. J. Fine and R. Harrop. Uniformization of linear arrays. *J. Symb. Logic*, 22:130–140, 1957.
- [10] F. Gardi. The Roberts characterization of proper and unit interval graphs. *Discrete Math.*, 307(22):2906–2908, 2007. doi:[10.1016/j.disc.2006.04.043](https://doi.org/10.1016/j.disc.2006.04.043).
- [11] M. C. Golumbic and P. L. Hammer. Stability in circular arc graphs. *J. Algorithms*, 9(3):314–320, 1988. doi:[10.1016/0196-6774\(88\)90023-5](https://doi.org/10.1016/0196-6774(88)90023-5).
- [12] N. Goodman. *The Structure of Appearance*. Boston Studies in the Philosophy and History of Science. Springer Netherlands, 3rd edition, 1977.
- [13] P. Hell and J. Huang. Certifying LexBFS recognition algorithms for proper interval graphs and proper interval bigraphs. *SIAM J. Discrete Math.*, 18(3):554–570 (electronic), 2004/05. doi:[10.1137/S0895480103430259](https://doi.org/10.1137/S0895480103430259).
- [14] J. Huang. On the structure of local tournaments. *J. Combin. Theory Ser. B*, 63(2):200–221, 1995. doi:[10.1006/jctb.1995.1016](https://doi.org/10.1006/jctb.1995.1016).
- [15] H. Kaplan and Y. Nussbaum. Certifying algorithms for recognizing proper circular-arc graphs and unit circular-arc graphs. *Discrete Appl. Math.*, 157(15):3216–3230, 2009. doi:[10.1016/j.dam.2009.07.002](https://doi.org/10.1016/j.dam.2009.07.002).
- [16] P. Klavík, J. Kratochvíl, and T. Vyskočil. Extending partial representations of interval graphs. In *Theory and applications of models of computation*, vol. 6648 of *Lecture Notes in Comput. Sci.*, pp. 276–285. Springer, Heidelberg, 2011. doi:[10.1007/978-3-642-20877-5_28](https://doi.org/10.1007/978-3-642-20877-5_28).
- [17] P. Klavík, J. Kratochvíl, T. Krawczyk, and B. Walczak. Extending partial representations of function graphs and permutation graphs. In *Algorithms—ESA 2012*, vol. 7501 of *Lecture Notes in Comput. Sci.*, pp. 671–682. Springer, Heidelberg, 2012. doi:[10.1007/978-3-642-33090-2_58](https://doi.org/10.1007/978-3-642-33090-2_58).
- [18] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. Extending partial representations of proper and unit interval graphs. *CoRR*, abs/1207.6960v2, 2014.
- [19] P. Klavík, J. Kratochvíl, Y. Otachi, I. Rutter, T. Saitoh, M. Saumell, and T. Vyskočil. Extending partial representations of proper and unit interval graphs. In R. Ravi and I. Gørtz, editors, *Algorithm Theory – SWAT 2014*, vol. 8503 of *Lecture Notes in Computer Science*, pp. 253–264. Springer, 2014. doi:[10.1007/978-3-319-08404-6_22](https://doi.org/10.1007/978-3-319-08404-6_22).
- [20] J. Köbler, S. Kuhnert, B. Laubner, and O. Verbitsky. Interval graphs: canonical representations in logspace. *SIAM J. Comput.*, 40(5):1292–1315, 2011. doi:[10.1137/10080395X](https://doi.org/10.1137/10080395X).
- [21] J. Köbler, S. Kuhnert, and O. Verbitsky. Solving the canonical representation and star system problems for proper circular-arc graphs in logspace. In *32nd International Conference on Foundations of Software Technology and Theoretical Computer Science*, vol. 18 of *LIPICs. Leibniz Int. Proc. Inform.*, pp. 387–399. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012.
- [22] J. Köbler, S. Kuhnert, and O. Verbitsky. Solving the canonical representation and star system problems for proper circular-arc graphs in log-space. *CoRR*, abs/1202.4406v5, 2013.

- [23] N. Limaye, M. Mahajan, and P. Nimbhorkar. Longest paths in planar DAGs in unambiguous log-space. *Chic. J. Theoret. Comput. Sci.*, pp. Article 8, 15, 2010.
- [24] M. C. Lin and J. L. Szwarcfiter. Unit circular-arc graph representations and feasible circulations. *SIAM J. Discrete Math.*, 22(1):409–423, 2008. doi:[10.1137/060650805](https://doi.org/10.1137/060650805).
- [25] M. C. Lin, F. J. Souignac, and J. L. Szwarcfiter. Short models for unit interval graphs. In *LAGOS'09—V Latin-American Algorithms, Graphs and Optimization Symposium*, vol. 35 of *Electron. Notes Discrete Math.*, pp. 247–255. Elsevier Sci. B. V., Amsterdam, 2009. doi:[10.1016/j.endm.2009.11.041](https://doi.org/10.1016/j.endm.2009.11.041).
- [26] M. C. Lin, D. Rautenbach, F. J. Souignac, and J. L. Szwarcfiter. Powers of cycles, powers of paths, and distance graphs. *Discrete Appl. Math.*, 159(7):621–627, 2011. doi:[10.1016/j.dam.2010.03.012](https://doi.org/10.1016/j.dam.2010.03.012).
- [27] J. Mitas. Minimal representation of semiorders with intervals of same length. In *Orders, algorithms, and applications (Lyon, 1994)*, vol. 831 of *Lecture Notes in Comput. Sci.*, pp. 162–175. Springer, Berlin, 1994. doi:[10.1007/BFb0019433](https://doi.org/10.1007/BFb0019433).
- [28] M. Pirlot. Minimal representation of a semiorder. *Theory and Decision*, 28(2):109–141, 1990. doi:[10.1007/BF00160932](https://doi.org/10.1007/BF00160932).
- [29] M. Pirlot. Synthetic description of a semiorder. *Discrete Appl. Math.*, 31(3):299–308, 1991. doi:[10.1016/0166-218X\(91\)90057-4](https://doi.org/10.1016/0166-218X(91)90057-4).
- [30] M. Pirlot and P. Vincke. *Semiorders*, vol. 36 of *Theory and Decision Library. Series B: Mathematical and Statistical Methods*. Kluwer Academic Publishers Group, Dordrecht, 1997. doi:[10.1007/978-94-015-8883-6](https://doi.org/10.1007/978-94-015-8883-6).
- [31] F. S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory (Proc. Second Ann Arbor Graph Theory Conf., Ann Arbor, Mich., 1968)*, pp. 139–146. Academic Press, New York, 1969.
- [32] F. J. Souignac. *On proper and Helly circular-arc graphs*. PhD thesis, Universidad de Buenos Aires, March 2010.
- [33] F. J. Souignac. Fully dynamic recognition of proper circular-arc graphs. *Algorithmica*, 2013. doi:[10.1007/s00453-013-9835-7](https://doi.org/10.1007/s00453-013-9835-7).
- [34] D. Stolee and N. V. Vinodchandran. Space-efficient algorithms for reachability in surface-embedded graphs. In *2012 IEEE 27th Conference on Computational Complexity—CCC 2012*, pp. 326–333. IEEE Computer Soc., Los Alamitos, CA, 2012. doi:[10.1109/CCC.2012.15](https://doi.org/10.1109/CCC.2012.15).
- [35] A. Tucker. Structure theorems for some circular-arc graphs. *Discrete Math.*, 7:167–195, 1974.